LEARNING WHAT NOT TO SAY: CONDITIONAL SEMANTIC CONTROL IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models can follow instructions about what to say, yet remain unreliable at controlling what not to say. We study *semantic omission*: completing a writing task while keeping an instructed semantic dimension absent from the generated text. Unlike surface-level constraints, semantic omission requires preventing the target semantics from reappearing through paraphrases, implicit references, and rhetorical structures, while preserving the utility of the remaining content. We construct a benchmark of 400 writing tasks across six domains and four writing settings, and systematically evaluate semantic omission across multiple model families. Our diagnosis reveals a behavioral gap between explicit boundary identification and artifact-level omission: models may correctly identify what should be excluded, yet still reintroduce the excluded semantics during generation. Thus, successful boundary identification alone is insufficient for reliable semantic omission. Motivated by this behavioral dissociation, we formulate the repair problem as *conditional semantic control* and propose SELECTOMIT, a post-training method that teaches models to selectively control whether a semantic dimension is expressed according to the current instruction. It jointly encourages suppression of the excluded semantics, preservation of the remaining task content, and recovery of the same semantics when explicitly requested. Experiments across multiple model families show that SELECTOMIT substantially improves semantic omission while preserving task utility, and enables reliable instruction-conditioned inclusion and exclusion of the same semantic content.

1 INTRODUCTION

Large language models (LLMs) have become capable writing assistants, generating fluent and task-appropriate text for academic writing, product descriptions, and professional communication (Raheja et al., 2023; Wu et al., 2025a). Yet effective writing assistance requires control over not only *what to say*, but also *what not to say*. For example, a user may ask for a product description without discussing its price; or a researcher may ask for a summary without discussing computational efficiency. The model must exclude the requested semantic dimension while preserving the rest of the intended content.

Semantic omission. We call this capability *semantic omission*: given a writing task and a semantic dimension to exclude, the model should complete the task while keeping that dimension absent from the generated text. Semantic omission is fundamentally different from keyword avoidance. The excluded semantics may reappear through paraphrases, implicit references, contrasts, or related expressions. Conversely, removing all content related to the target dimension can make the response incomplete or unhelpful. Successful omission therefore requires *selectively excluding one semantic dimension while preserving the remaining task-relevant content*. Consider the instruction, "describe the product without discussing price." A model may respond with 'Although price is not the main advantage, ...", explicitly mentioning the very dimension that the user intended to exclude. More subtly, it may avoid the word "price" while still discussing related concepts such as affordability, cost, or value for money. These failures raise the possibility that a model may be able to explicitly identify what should be excluded yet still fail to maintain that exclusion during generation. This raises a broader question: *can language models conditionally control whether a semantic dimension appears in their generated text according to the current instruction?*

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

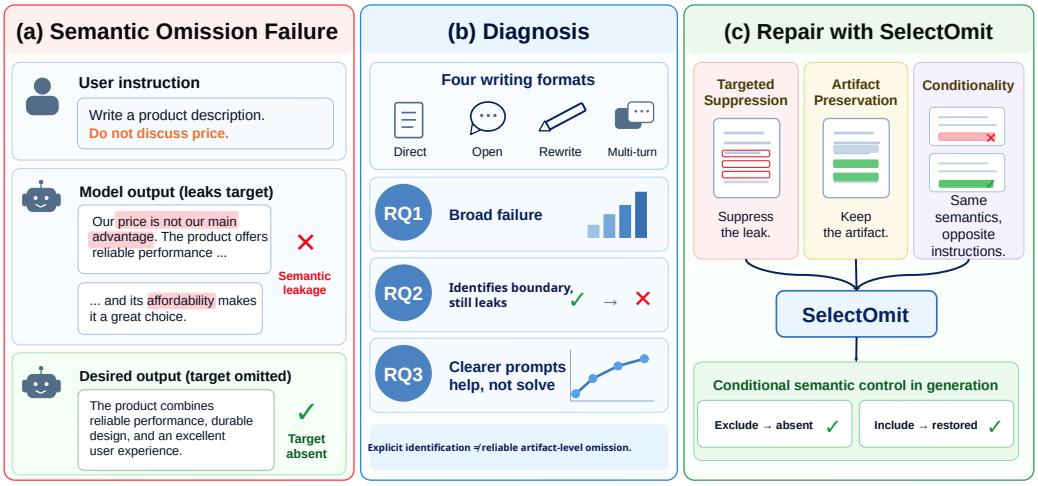

Figure 1: Overview of semantic omission and our approach. (a) Models may reintroduce an excluded semantic dimension through direct or indirect expressions. (b) Our diagnosis shows that successful explicit boundary identification and clearer prompting do not guarantee artifact-level omission. (c) SELECTOMIT jointly learns targeted suppression, artifact preservation, and instruction-conditioned recovery to achieve conditional semantic control.

From boundary identification to semantic control. Instruction-following research has extensively studied explicit constraints on format, length, style, and content (Zhou et al., 2023; Jiang et al., 2024; Zhang et al., 2025; Ren et al., 2025), while constrained decoding can enforce lexical or rule-based restrictions (Lu et al., 2021; Jin & Han, 2026). Such constraints are often directly verifiable from surface forms. In contrast, semantic omission requires controlling a context-dependent region of meaning: the same semantic dimension may need to be included under one instruction and excluded under another. Prior work on negative instructions and negation has studied whether language models correctly interpret negated requests and how negation affects model behavior (Jang et al., 2022; She et al., 2023; Rezaei & Blanco, 2024; Mann et al., 2026). Our focus is complementary: rather than asking only whether a model can identify the intended scope of a negative instruction, we ask whether it can *maintain the requested semantic exclusion while completing the underlying task*. This distinction motivates us to separate two observable behaviors: explicit boundary identification and artifact-level omission. As we show empirically, providing an explicit exclusion boundary or asking models to restate the constraint improves performance, but substantial semantic leakage remains.

Diagnosing semantic omission. To study this problem systematically, we construct a benchmark of 100 matched scenarios across six writing domains, instantiated in four settings that vary in where the excluded semantics originate: provided source material, an open-ended task, an existing draft, and prior conversational context. This yields 400 writing tasks, each specifying a semantic dimension to exclude and the content to preserve. We evaluate both *semantic omission* and *task completion*. Across five checkpoints from four model families, we observe substantial and structured failures. For example, Qwen3-4B satisfies both requirements on only 80.42% of examples overall, dropping to 47.67% in the Rewrite Removal setting, where the target semantics are already instantiated in the draft. Importantly, the failures persist even when the exclusion is explicitly identified. These results show that semantic omission cannot be fully explained by failures of explicit boundary identification: even when the intended boundary is correctly identified, it may still be violated in the final artifact.

Learning conditional semantic control. This diagnosis suggests that omission should not be learned as global suppression. A model should omit "price" when instructed not to discuss it, yet naturally discuss price when explicitly asked to do so, without sacrificing the rest of the requested artifact. We therefore formulate the repair target as *conditional semantic control* and propose SELECTOMIT, a single-stage post-training method with three complementary training signals: (i) It applies token-level unlikelihood only to the spans that actually invoke the excluded semantics, rather than suppressing an otherwise useful response. (ii) It combines clean-response supervision with a util-

ity preference that favors complete artifacts over omission achieved through over-deletion, brevity, genericity, or refusal. (iii) It constructs counterfactual inclusion instructions for the same semantic targets and explicitly trains the model to restore those semantics when the instruction changes. These signals are optimized jointly.

Experiments show substantial improvements in both omission and conditional control. On Qwen3-4B, the proportion of examples satisfying both omission and task completion increases from 80.00% to 97.50%. Under paired instructions requiring the same semantic dimension to be excluded or included, SELECTOMIT satisfies both instructions in 81.50% of paired tasks. With two semantic dimensions and all four include/exclude combinations, it succeeds on all four conditions for 74% of scenarios. Improvements across multiple model families further support that SELECTOMIT learns instruction-dependent semantic control rather than merely suppressing particular concepts. Figure 1 summarizes the semantic omission failure, our diagnostic analysis, and the resulting repair strategy.

2 SEMANTIC OMISSION: TASK AND EVALUATION FRAMEWORK

We formalize *semantic omission* as a conditional generation problem in which a language model must produce a useful writing artifact while excluding a designated semantic dimension. The key challenge is to suppress the excluded semantics without unnecessarily removing or degrading the retained content. We therefore evaluate two requirements separately: whether the excluded semantics remain absent, and whether the requested artifact remains complete and useful. Section 3 uses this framework to diagnose current language models, while Section 4 studies post-training repair.

2.1 TASK FORMULATION

We represent a writing task as $x = (w, c, e, k, a)$, where w is the writing request, c denotes optional source material or dialogue context, e is the semantic dimension to exclude, k specifies the content that should be retained, and a contains additional artifact requirements such as length, style, or format. Given a candidate output y, we define two binary criteria: $O(y; x) \in \{0, 1\}$ indicates whether the excluded semantic dimension is absent, and $T(y; x) \in \{0, 1\}$ indicates whether the requested writing task is fulfilled. We define successful semantic omission as satisfying both requirements:

$$V(y; x) = O(y; x) \cdot T(y; x). \tag{1}$$

Crucially, omission is defined at the semantic level rather than by lexical matching. An output violates omission whenever its interpreted meaning invokes the excluded dimension, even if the target is expressed through paraphrase, implication, polarity reversal, contrast, qualification, or other discourse structures. Conversely, lexical overlap with the excluded vocabulary does not constitute leakage when the excluded semantics are not invoked. Task fulfillment requires the requested artifact form, topical relevance, coherent and substantive content, and sufficient coverage of k.

2.2 BENCHMARK CONSTRUCTION

We build the benchmark with a structured LLM-assisted construction pipeline. First, candidate base scenarios are generated under a fixed schema specifying the writing goal, domain, excluded semantic dimension e, retained content k, difficulty, and artifact requirements. Candidates are then checked for required fields, scenario consistency, feasibility after omission, and near-duplicates, followed by manual review of semantic clarity and whether the task remains meaningful without e. This process yields 100 retained base scenarios spanning six writing domains: academic writing, product and marketing copy, professional writing, reports and analysis, public communication, and reviews and recommendations.

Each retained scenario is instantiated with four fixed task templates that vary how the excluded semantics enter the context: **Direct Exclusion**, **Open-ended Scope Exclusion**, **Rewrite Removal**, and **Multi-turn Update**, yielding 400 English writing tasks. Appendix A.2 gives the candidate-generation prompt schema, the four task templates, and representative construction details; Appendix A.1 reports benchmark statistics and quality checks. The complete frozen benchmark is designated for release as a supplementary JSONL artifact.

We further annotate representative leakage patterns, including direct mention, semantic paraphrase, polarity reversal, identifiable implication, and rhetorical leakage through contrast, qualification,

meta-commentary, or discourse organization. These annotations support fine-grained diagnosis, while the primary omission criterion remains the artifact-level question of whether the excluded semantic dimension is meaningfully invoked.

2.3 Evaluation Setup

Models and Generation Settings. We evaluate Qwen3-4B and Qwen3-8B (Qwen Team, 2025), Llama-3.1-8B-Instruct (Llama Team, 2024), GLM-4-9B-Chat (GLM Team, 2024), and Gemma-2-9B-it (Gemma Team, 2024), covering five checkpoints from four model families. Each model independently generates three responses per task using temperature 0.7, top-p 0.9, a 384-token generation budget, and a 4096-token input limit.

Evaluation. Because semantic omission cannot be reliably assessed by string matching, we use GPT-5.6 Sol as a fixed rubric-based evaluator, following prior use of strong LLMs and explicit rubrics for open-ended generation evaluation (Zheng et al., 2023; Wu et al., 2025a; He et al., 2026). The evaluator separately judges semantic omission and task fulfillment from the original task, excluded boundary, retained-content specification, and full candidate artifact. It checks direct and indirect leakage forms and requires supporting evidence from the response.

All experiments use the same evaluation criteria and protocol. Appendix A.3 provides the complete rubric and judge prompt, and Appendix A.4 reports independent human validation. Diagnostic prompts and additional diagnostic results are collected in Appendix B, while repair protocols and supplementary repair analyses are collected in Appendix C.

3 Diagnosing Semantic Omission

We diagnose semantic omission from three complementary perspectives: *failure prevalence*, *boundary awareness*, and *boundary control*. We first characterize where omission fails, then examine whether models can identify the intended semantic boundary and maintain it during generation. The results reveal a consistent behavioral gap between *knowing what to exclude* and *actually excluding it*, motivating the post-training approach in Section 4.

3.1 RQ1: Is Semantic Omission Failure Prevalent?

We evaluate five checkpoints on all 400 tasks. Table 1 shows that semantic omission failures occur across all model families, with overall valid success ranging from 80.42% to 95.25%. More importantly, performance varies substantially with how the excluded semantics are presented. *Rewrite Removal* is consistently the most challenging setting for four of the five models: when the excluded semantics are already expressed in the input artifact, valid success drops to 47.67% for Qwen3-4B, 68.33% for Qwen3-8B, and 68.67% for GLM-4-9B, while Llama and Gemma are more robust. These results establish two properties of semantic omission: it is a *cross-model* failure rather than a model-specific artifact, and it becomes substantially harder when the excluded semantics are already salient in the artifact being generated or revised. This motivates a deeper question: *do models fail because they cannot identify the semantic boundary, or because they cannot maintain it during generation?* Appendix B.3 repeats the diagnosis on stronger API-accessible models.

Table 1: Valid success (%) across the four semantic omission writing formats.

Model	Direct	Open	Rewrite	Multi-turn	Overall
Qwen3-4B	88.00	90.67	47.67	95.33	80.42
Qwen3-8B	89.33	92.33	68.33	96.67	86.67
Llama-3.1-8B	90.00	88.00	88.00	98.33	91.08
GLM-4-9B	81.33	90.67	68.67	92.33	83.25
Gemma-2-9B	94.67	95.00	95.00	96.33	95.25

3.2 RQ2: Does Knowing the Boundary Ensure Omission?

A natural explanation for RQ1 is that models may simply fail to identify what the instruction requires them to exclude. We therefore separately probe whether a model can explicitly articulate the intended exclusion boundary and whether the subsequent artifact respects that boundary. The probe prompt, scoring protocol, and exact identification counts are reported in Appendix B.1. Figure 2 reveals a clear behavioral dissociation. Even when the probe correctly identifies the intended boundary, semantic leakage persists in 17.99% of Qwen3-4B responses, 11.73% of Qwen3-8B responses, 7.68% of Llama responses, 12.41% of GLM responses, and 3.36% of Gemma responses. Thus, knowing what to exclude does not guarantee excluding it during generation. The failure cannot be reduced to a lack of explicit boundary awareness: models can represent the correct semantic boundary yet reintroduce the excluded semantics in the final artifact.

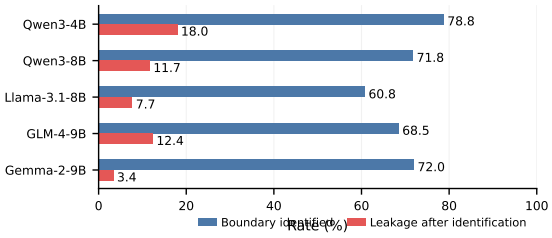

Figure 2: Among responses whose exclusion boundary is correctly identified, a non-trivial fraction still exhibits semantic leakage.

3.3 RQ3: Can Clearer Instructions Solve Semantic Omission?

We next ask whether the remaining failures are caused by ambiguity in the original instruction. We progressively strengthen the specification through four interventions: *Strong* explicitly prohibits indirect and rhetorical references; *Self* asks the model to restate the boundary before writing; *Oracle* directly provides the exclusion boundary and retained focus; and *Focus-only* specifies only what should remain. Exact intervention prompts and additional results are provided in Appendix B.2. Clarifying the instruction improves performance, but does not eliminate semantic leakage. In particular, even the *Oracle* intervention, which removes uncertainty about the intended boundary, still leaves omission failures of 6.08%, 2.67%, and 3.08% for Qwen3-4B, Qwen3-8B, and Llama, respectively. Positive *Focus-only* guidance also produces a trade-off between omission and task fulfillment: for Llama, omission success increases from 91.58% to 96.83%, while task fulfillment decreases from 99.33% to 92.67%. These results show that better boundary specification reduces ambiguity but does not guarantee boundary adherence during generation. The core bottleneck is therefore not identifying or specifying the semantic boundary, but controlling generation conditioned on that boundary. This motivates our post-training approach, which directly trains models to selectively suppress or express the same semantic content according to the current instruction while preserving task utility.

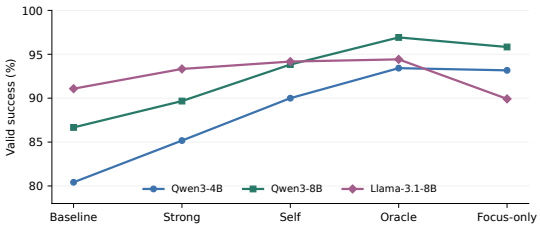

Figure 3: Valid success under increasingly explicit boundary interventions.

4 Repairing Semantic Omission

Our diagnosis shows that semantic omission is not primarily a boundary-identification problem, but a generation-control problem. We therefore propose SelectOmit, a post-training framework that learns three complementary behaviors: *Suppress* the excluded semantics, *Preserve* the remaining artifact, and *Recover* the same semantics when the instruction changes. Figure 1 summarizes the approach.

4.1 SelectOmit

Given an exclusion instruction x_i^- with target semantic dimension e, SelectOmit constructs complementary supervision for suppression, preservation, and conditional recovery, and optimizes them jointly in a single post-training stage.

Suppress: Localized Semantic Suppression. A leaking response y_i^ℓ may be largely correct except for a small amount of content that invokes e. We therefore annotate the leaked token positions S_i^- and apply token-level unlikelihood (Welleck et al., 2019) only to these positions:

$$\mathcal{L}_{\text{omit},i} = -\frac{1}{|S_i^-|} \sum_{t \in S_i^-} \log\big(1 - p_\theta(y_{i,t}^\ell \mid x_i^-, y_{i,<t}^\ell)\big)\,. \tag{2}$$

Unlike response-level negative supervision, this localized objective suppresses the semantic leakage while leaving the rest of the response unconstrained.

Preserve: Useful Artifact Generation. For the same exclusion instruction, we construct a clean response y_i^c that omits e while fully completing the writing task, and train it with standard cross-entropy,

$$\mathcal{L}_{\text{clean},i} = -\frac{1}{|y_i^c|} \sum_t \log p_\theta(y_{i,t}^c \mid x_i^-, y_{i,<t}^c). \tag{3}$$

To distinguish useful omission from degenerate avoidance, we additionally construct a utility-negative response y_i^u that omits e but suffers from over-deletion, brevity, genericity, undercoverage, or partial refusal. We prefer y_i^c over y_i^u with a DPO-style objective (Rafailov et al., 2023):

$$\mathcal{L}_{\text{utility},i} = -\log \sigma\big(\beta[r_\theta(x_i^-, y_i^c) - r_\theta(x_i^-, y_i^u)]\big)\,, \quad r_\theta(x,y) = \log \frac{p_\theta(y \mid x)}{p_{\text{ref}}(y \mid x)}. \tag{4}$$

Because both candidates satisfy omission, this preference specifically optimizes artifact utility rather than omission alone.

Recover: Conditional Semantic Control. For a subset of groups, we keep the scenario and semantic target fixed but replace the exclusion prompt x_i^- with an inclusion prompt x_i^+ that explicitly requests the same dimension. We provide an inclusion response y_i^+ and apply token-level cross-entropy only to the annotated target spans S_i^+:

$$\mathcal{L}_{\text{include},i} = -\frac{1}{|S_i^+|} \sum_{t \in S_i^+} \log p_\theta(y_{i,t}^+ \mid x_i^+, y_{i,<t}^+). \tag{5}$$

The exclusion and inclusion examples therefore provide opposite supervision for the same semantic content: $x_i^- : e \to \text{omit}$ and $x_i^+ : e \to \text{include}$. The goal is not to increase inclusion generally, but to ensure that suppression is reversible when the instruction changes. This discourages global aversion to e and instead binds semantic expression to the current instruction.

Joint Optimization. Rather than training these objectives sequentially, SELECTOMIT jointly samples their supervision in one post-training process. Each training group contributes the losses supported by its available annotations, and all components update the same model:

$$\mathcal{L}_i = \lambda_o \mathcal{L}_{\text{omit},i} + \lambda_c \mathcal{L}_{\text{clean},i} + \lambda_u \mathcal{L}_{\text{utility},i} + \lambda_+ \mathcal{L}_{\text{include},i}. \tag{6}$$

The coefficients λ_o, λ_c, λ_u, and λ_+ balance the four training objectives; their settings and the complete optimization configuration are reported in Appendix C.4. Training-data construction is detailed in Appendix C.1.

4.2 EXPERIMENTS

We use Qwen3-4B as the primary model and evaluate on the held-out benchmark. Because semantic omission is introduced here as a new artifact-level writing task and no prior method directly targets this instruction-conditioned setting, we construct controlled comparison methods from standard post-training paradigms. We compare the unadapted Base model with post-training variants: (i) Matched SFT on clean responses, (ii) Mixed SFT with additional boundary-identification supervision, (iii) Positive-matched SFT with the same inclusion responses available to SELECTOMIT, (iv) Span-matched Conditional SFT with localized inclusion-span supervision, and (v) Standard DPO on clean-versus-leaking response pairs. Exact training data and optimization recipes for all baselines are given in Appendix C.4.1. We further test transfer to other model families, instruction switching, compositional control, and component ablations; the corresponding supplementary analyses and uncertainty estimates are collected in Appendix C.

4.2.1 CAN SELECTOMIT EFFECTIVELY REPAIR SEMANTIC OMISSION?

Table 2 shows that valid success rises from 80.00% for Base to 97.50% for SELECTOMIT, compared with 93.25–93.50% for the SFT variants and 96.50% for DPO. Training and baseline implementation details are reported in Appendices C.4 and C.4.1. Thus targeted post-training can repair semantic omission without sacrificing task fulfillment. Importantly, DPO is already strong in the exclusion-only setting, so omission accuracy alone cannot distinguish selective silence from broader target suppression; the next experiments test that distinction directly.

To test whether the repair is specific to one checkpoint, we apply the same SELECTOMIT recipe to Qwen3-8B, Llama-3.1-8B, and Gemma-2-9B. As shown in Table 3, all three improve, with gains of 11.75, 8.75, and 1.75 points, respectively. The smaller Gemma gain is consistent with its already high starting performance. These results suggest that the repair transfers across model families and initial capability levels.

Table 3: Cross-model valid success (%).

Model	Base	Ours	Δ
Qwen3-4B	80.00	97.50	+17.50
Qwen3-8B	85.75	97.50	+11.75
Llama-3.1-8B	89.25	98.00	+8.75
Gemma-2-9B	95.50	97.25	+1.75

4.2.2 DOES SELECTOMIT TRULY LEARN CONDITIONAL SEMANTIC CONTROL?

Instruction Switching. For each semantic target, we pair the original exclusion instruction with an inclusion instruction that explicitly requests the same content. A pair succeeds only if both artifacts are correct. The exact switching/compositional evaluation protocol and blinded human method comparison are reported in Appendices C.5 and C.7. Table 4 exposes the key difference hidden by exclusion-only evaluation. Standard DPO reaches 95.75% exclusion validity but only 64.50% inclusion validity, yielding 62.75% paired validity, showing that strong exclusion can reflect broader suppression of the target semantics. Importantly, Span-matched Conditional SFT controls for the matched exclusion/inclusion data and localized inclusion-span supervision used by SELECTOMIT, yet reaches only 74.00% paired validity. Thus, the improvement of SELECTOMIT cannot be explained by counterfactual inclusion data or span-localized supervision alone. SELECTOMIT achieves 96.75% exclusion validity and 83.75% inclusion validity, resulting in 81.50% paired validity and more reliably conditioning whether the target semantics are expressed on the current instruction.

Compositional Semantic Control. We further select two semantic dimensions A and B per scenario and evaluate (A^+, B^+), (A^-, B^+), (A^+, B^-), and (A^-, B^-). A scenario succeeds only when all four independently generated artifacts are valid. As Figure 4 shows, SELECTOMIT reaches 74% all-four validity, compared with 58% for Base, 59% for Matched SFT, 37% for DPO, and 52% for Positive-matched SFT. DPO remains strong when both dimensions are excluded but degrades sharply when both must be included, whereas SELECTOMIT is more balanced across all four combinations. The gap therefore reflects instruction-dependent control over multiple semantic dimensions rather than simply stronger suppression. We additionally verify the same behavior at the likelihood level: inclusion supervision primarily restores target likelihood under inclusion instructions while largely preserving exclusion-time suppression. The full conditional-control protocol and likelihood-level analysis are reported in Appendices C.5 and C.6; transfer to authentic source-grounded writing is evaluated in Appendix C.9.

4.2.3 HOW DO THE TRAINING OBJECTIVES CONTRIBUTE?

Table 5 reports the component ablations, which support the three design goals. Removing span-level unlikelihood reduces valid success from 97.50% to 93.25%, while response-wide suppression reaches 94.50%, supporting localized negative supervision. Removing clean-response CE leaves omission high at 98.75% but collapses task fulfillment to 33.50%, showing that suppression alone can degenerate into unhelpful avoidance. The utility preference provides an additional gain in artifact quality. Finally, removing inclusion supervision has little effect on

Table 5: Ablation results (%) on the held-out benchmark.

Variant	Omit.	Task	Valid
SELECTOMIT	97.75	99.75	**97.50**
w/o Span UL	93.50	99.75	93.25
Resp.-wide UL	96.25	97.50	94.50
w/o Clean CE	98.75	33.50	33.50
w/o Utility	97.25	99.00	96.25
w/o Inclusion	97.00	99.25	96.50

Table 2: Held-out semantic omission results (%) on Qwen3-4B. Valid success requires both successful omission and fulfillment of the original writing task.

Method	Omission	Task Fulfillment	Valid Success
Base	80.50	98.75	80.00
Matched SFT	93.50	99.50	93.25
Mixed SFT	93.75	99.25	93.50
Positive-matched SFT	93.75	99.50	93.25
Standard DPO	97.75	98.75	96.50
SELECTOMIT	97.75	99.75	**97.50**

Table 4: Held-out instruction switching (%). Paired validity requires both the exclusion and inclusion conditions to pass for the same semantic target.

Method	Exclude V	Include V	Paired V
Base	80.75	92.25	74.00
Matched SFT	94.00	78.25	73.25
Positive-matched SFT	93.25	76.25	71.25
Span-matched Conditional SFT	94.00	78.50	74.00
Standard DPO	95.75	64.50	62.75
w/o Inclusion CE	96.75	73.25	71.25
SELECTOMIT	96.75	83.75	**81.50**

exclusion-only performance but reduces inclusion validity from 83.75% to 73.25% and paired validity from 81.50% to 71.25%. Overall, the three objectives play distinct roles: *Suppress* the target, *Preserve* the artifact, and *Recover* the target when the instruction changes.

5 RELATED WORK

Negation and Negative Instructions. Prior work has shown that negation remains challenging for language models. Jang et al. (Jang et al., 2022) find that simply reformulating an instruction into a negated form can substantially degrade performance, while ScoNe (She et al., 2023) shows that model behavior is highly sensitive to negation scope and context. Subsequent work explores how to improve negation understanding: Rezaei and Blanco (Rezaei & Blanco, 2024) show that rewriting negated inputs into affirmative interpretations improves performance on downstream tasks, while *Commonsense Knowledge with Negation* (Wang et al., 2026) constructs large-scale negation-augmented commonsense data and shows that targeted training can strengthen negation understanding. More recently, *Don't Think of the White Bear* (Mann et al., 2026) studies "do not mention X" instructions and identifies an ironic rebound effect, where suppressing a concept can instead make it more salient.

These studies mainly examine how negation is understood, represented, or suppressed. Our work focuses on a related but more specific problem in open-ended writing: *selective semantic exclusion*. The model must keep a designated semantic dimension absent while preserving the rest of a useful writing artifact. We study whether an excluded dimension can still re-enter the final text through paraphrases, implicit references, contrasts, or rhetorical framing, even when the model can explicitly identify that it should be excluded.

Instruction and Constraint Following. Semantic omission can also be viewed as a special case of *instruction and constraint following*, where model outputs must satisfy user-specified requirements. Existing benchmarks have progressively expanded the scope of such evaluation: IFEval (Zhou et al., 2023) focuses on constraints that can be reliably checked with rules; FollowBench (Jiang et al., 2024) organizes constraints into categories such as content, situation, style, format, and example; ComplexBench (Wen et al., 2024) studies the composition of multiple constraints; CFBench (Zhang et al., 2025) introduces richer constraints from realistic user scenarios; MulDimIF (Ye et al., 2026) evaluates instruction following across multiple constraint patterns, categories, and difficulty levels; and AdvancedIF (He et al., 2026) further uses expert-designed rubrics to evaluate more open-ended, multi-turn, and semantic constraints.

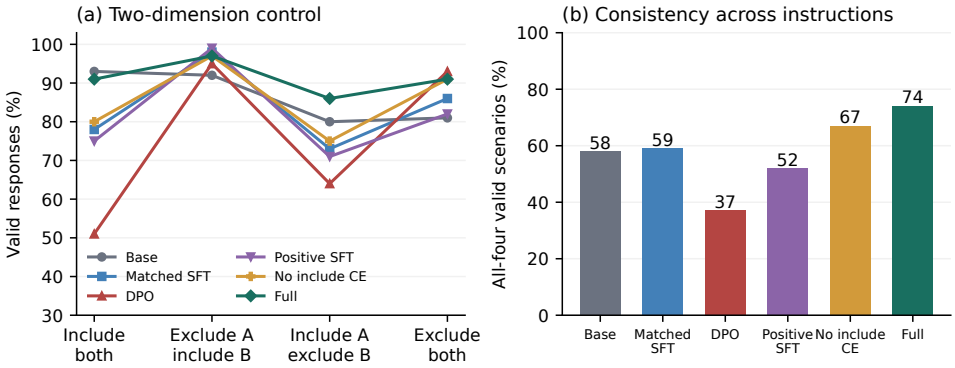

Figure 4: Compositional semantic control over two dimensions.

Methods for improving constraint following generally intervene at three stages. (i) *Instruction reformulation and decomposition* rewrites, decomposes, or explicitly restates constraints to make them easier to follow, as in affirmative reformulation for negation (Rezaei & Blanco, 2024) and decomposition–critique–refinement pipelines such as DeCRIM (Ferraz et al., 2024); we similarly test stronger prohibitions, self-restated boundaries, and directly provided semantic boundaries. (ii) *Constrained decoding* directly restricts generation, as in NeuroLogic Decoding (Lu et al., 2021) and NCO (Jin & Han, 2026), but such methods are most suitable when constraints can be represented as lexical, logical, or finite patterns; semantic omission instead requires suppressing a context-dependent semantic dimension that may reappear through paraphrase or discourse structure. (iii) *Post-training* improves constraint following through supervised fine-tuning on constraint-rich data (Ye et al., 2026), preference optimization for semantically defined soft constraints (Ren et al., 2025; Rafailov et al., 2023), or reinforcement learning with constraint-wise or rubric-based rewards (He et al., 2026; Ren et al., 2026). Our work follows this direction, but addresses an additional challenge specific to semantic omission: the model must learn to exclude a semantic dimension when requested without turning that exclusion into permanent suppression of the same content.

Concept Steering. A closely related line of work studies fine-grained control over whether particular semantic concepts are expressed in language-model generations. AxBench (Wu et al., 2025b) provides a large-scale benchmark for concept detection and model steering, comparing prompting, finetuning, and representation-based interventions. More recently, RePS (Wu et al., 2025c) introduces a bidirectional preference objective for representation steering that supports both concept promotion and concept suppression. These studies establish that targeted semantic content can be actively increased or suppressed through prompting, parameter updates, or representation interventions. Existing concept-steering methods modulate target concepts through explicit control mechanisms such as prompting, parameter adaptation, or inference-time representation interventions. Our setting instead asks whether an ordinary natural-language writing instruction can induce the same model to selectively omit or restore a semantic dimension while preserving the rest of a complete artifact. We therefore view semantic omission as complementary to concept steering: the former studies instruction-conditioned artifact-level control, while the latter studies direct interventions for modulating concept expression.

6 Conclusion

We study *semantic omission*, the ability of language models to keep a user-excluded semantic dimension out of a complete writing artifact while preserving the rest of the task. Our diagnosis shows that this failure appears across model families and writing settings, persists even after successful explicit boundary identification, and remains under increasingly explicit boundary specification. To repair this behavior, we formulate the problem as *conditional semantic control* and introduce SelectOmit, which jointly learns targeted suppression, artifact preservation, and instruction-dependent restoration of the same semantic content. Experiments show that SelectOmit substantially improves semantic omission across models while avoiding the failure mode of unconditional suppression, leading to stronger include/exclude switching and compositional control. These results suggest that reliable selective silence requires artifact-level control over when a semantic dimension should remain absent and when it should reappear under the current instruction.

Reproducibility Statement

We provide the benchmark definition, evaluation protocols, training-data construction procedures, and implementation details needed to reproduce the main experiments. Section 2 describes the semantic omission task formulation and evaluation framework. Appendix A reports benchmark construction prompts, task templates, statistics, the evaluation rubric, and evaluator validation; Appendix B provides the diagnostic probe/intervention protocols and additional diagnosis on larger models; and Appendix C collects SELECTOMIT training-data construction, optimization and baseline implementations, conditional-control evaluations, supplementary repair experiments, and uncertainty analyses.

AI Use Statement

Generative AI tools were used at multiple stages of this research. Specifically, large language models were used to (i) assist in constructing and filtering parts of the synthetic benchmark and training data; (ii) perform rubric-based automatic evaluation; (iii) assist with writing and debugging experimental code; and (iv) support drafting and polishing the manuscript. The main automatic evaluation reported in this paper uses a fixed GPT-5.6 Sol evaluator with a unified rubric for semantic omission and task fulfillment. AI-generated or AI-assisted data were subjected to programmatic checks and human validation. The key experimental design choices, evaluation criteria, result analyses, and final conclusions presented in the paper were determined and reviewed by the authors. Generative AI was not used as a substitute for author judgment regarding the research findings, and the authors take full responsibility for the manuscript, experimental results, and all AI-assisted research artifacts.

References

Thomas Palmeira Ferraz, Kartik Mehta, Yu-Hsiang Lin, Haw-Shiuan Chang, Shereen Oraby, Sijia Liu, Vivek Subramanian, Tagyoung Chung, Mohit Bansal, and Nanyun Peng. LLM self-correction with DeCRIM: Decompose, critique, and refine for enhanced following of instructions with multiple constraints. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 7773–7812. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.458. URL https://aclanthology.org/2024.findings-emnlp.458/.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024. URL https://arxiv.org/abs/2408.00118.

GLM Team. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024. URL https://arxiv.org/abs/2406.12793.

Yun He, Wenzhe Li, Hejia Zhang, Songlin Li, Karishma Mandyam, Sopan Khosla, Yuanhao Xiong, Nanshu Wang, Xiaoliang Peng, Beibin Li, Shengjie Bi, Shishir G. Patil, Qi Qi, Shengyu Feng, Julian Katz-Samuels, Richard Yuanzhe Pang, Sujan Kumar Gonugondla, Hunter Lang, Yue Yu, Yundi Qian, Maryam Fazel-Zarandi, Licheng Yu, Amine Benhalloum, Hany Hassan Awadalla, and Manaal Faruqui. AdvancedIF: Rubric-based benchmarking and reinforcement learning for advancing LLM instruction following. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18003–18022. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.820. URL https://aclanthology.org/2026.acl-long.820/.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Joel Jang, Seonghyeon Ye, and Minjoon Seo. Can large language models truly understand prompts? a case study with negated prompts. *arXiv preprint arXiv:2209.12711*, 2022. URL https://arxiv.org/abs/2209.12711.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. FollowBench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4667–4688. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.257. URL https://aclanthology.org/2024.acl-long.257/.

Hyundong Jin and Yo-Sub Han. NCO: A versatile plug-in for handling negative constraints in decoding. *arXiv preprint arXiv:2605.10065*, 2026. URL https://arxiv.org/abs/2605.10065.

Llama Team. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ximing Lu, Peter West, Rowan Zellers, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. NeuroLogic decoding: (un)supervised neural text generation with predicate logic constraints. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4288–4299. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.339. URL https://aclanthology.org/2021.naacl-main.339/.

Logan Mann, Nayan Saxena, Sarah Tandon, Chenhao Sun, Savar Toteja, and Kevin Zhu. Don't think of the white bear: Ironic negation in transformer models under cognitive load. In *Proceedings of the 48th Annual Meeting of the Cognitive Science Society*. Cognitive Science Society, 2026. URL https://escholarship.org/uc/item/0zs8h426.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023. URL https://arxiv.org/abs/2305.18290.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5274–5291, 2023. URL https://aclanthology.org/2023.findings-emnlp.350/.

Qingyu Ren, Jie Zeng, Qianyu He, Jiaqing Liang, Yanghua Xiao, Weikang Zhou, Zeye Sun, and Fei Yu. Step-by-step mastery: Enhancing soft constraint following ability of large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19581–19596. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.1004. URL https://aclanthology.org/2025.findings-acl.1004/.

Qingyu Ren, Qianyu He, Powei Chang, Jie Zeng, Zeye Sun, Fei Yu, Jiaqing Liang, and Yanghua Xiao. Instructions are all you need: Self-supervised reinforcement learning for instruction following. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4755–4776. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.217. URL https://aclanthology.org/2026.acl-long.217/.

MohammadHossein Rezaei and Eduardo Blanco. Paraphrasing in affirmative terms improves negation understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 602–615. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-short.55. URL https://aclanthology.org/2024.acl-short.55/.

Jingyuan S. She, Christopher Potts, Samuel R. Bowman, and Atticus Geiger. Scone: Benchmarking negation reasoning in language models with fine-tuning and in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2023. URL https://aclanthology.org/2023.acl-short.154/.

Zijie Wang, MohammadHossein Rezaei, Farzana Rashid, and Eduardo Blanco. Commonsense knowledge with negation: A resource to enhance negation understanding. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11915–11934. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.578. URL https://aclanthology.org/2026.findings-acl.578/.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *arXiv preprint arXiv:1908.04319*, 2019. URL https://arxiv.org/abs/1908.04319.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmarking complex instruction-following with multiple constraints composition. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-4371. URL https://arxiv.org/abs/2407.03978.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. Writingbench: A comprehensive benchmark for generative writing. In *Advances in Neural Information Processing Systems*, volume 38, 2025a. doi: 10.52202/085713-1737. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/4aedf0cba303537fcb6cf948bb41b2df-Abstract-Datasets_and_Benchmarks_Track.html.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67035–67080. PMLR, 2025b. URL https://proceedings.mlr.press/v267/wu25a.html.

Zhengxuan Wu, Qinan Yu, Aryaman Arora, Christopher D. Manning, and Christopher Potts. Improved representation steering for language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025c. doi: 10.52202/085713-5364. URL https://proceedings.neurips.cc/paper_files/

paper/2025/hash/eb1ef82926376d252dde00d5dd909f4b-Abstract-Conference.html.

Junjie Ye, Caishuang Huang, Zhuohan Chen, Wenjie Fu, Chenyuan Yang, Leyi Yang, Yilong Wu, Peng Wang, Meng Zhou, Xiaolong Yang, Tao Gui, Qi Zhang, Zhongchao Shi, Jianping Fan, and Xuanjing Huang. MulDimIF: A multi-dimensional constraint framework for evaluating and improving instruction following in large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 2078–2104. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.99. URL https://aclanthology.org/2026.findings-acl.99/.

Tao Zhang, ChengLIn Zhu, Yanjun Shen, Wenjing Luo, Yan Zhang, Hao Liang, Tao Zhang, Fan Yang, Mingan Lin, Yujing Qiao, Weipeng Chen, Bin Cui, Wentao Zhang, and Zenan Zhou. CFBench: A comprehensive constraints-following benchmark for LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32926–32944. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1581. URL https://aclanthology.org/2025.acl-long.1581/.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023. URL https://arxiv.org/abs/2306.05685.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023. URL https://arxiv.org/abs/2311.07911.

A Benchmark and Evaluation Details

A.1 Benchmark Construction and Statistics

The diagnostic benchmark contains 100 base scenarios, each instantiated in the four writing formats introduced in Section 2.2. The scenarios span six writing domains: 20 academic-writing scenarios, 17 product/marketing scenarios, 17 professional-writing scenarios, 17 reports/analysis scenarios, 13 public-communication scenarios, and 16 reviews/recommendations scenarios. Together, the four variants produce 400 benchmark items.

The benchmark is generated from structured scenario specifications. Each specification defines an excluded semantic boundary and the content that should remain after exclusion. Mechanical validation checks required fields, scenario consistency, and near duplicates across scenario groups.

Table 6: Difficulty distribution of the diagnostic benchmark.

Task format	Easy	Medium	Hard
Direct Exclusion	40	40	20
Open-ended Scope Exclusion	15	45	40
Rewrite Removal	25	40	35
Multi-turn Update	20	40	40

A.2 Benchmark Construction Prompt and Task Templates

We use an LLM-assisted candidate-generation pipeline with a fixed structured schema. The generator proposes realistic writing scenarios rather than lexical "banned-word" tasks. Each candidate record specifies the domain, writing goal, excluded semantic dimension, retained-content requirements, difficulty, and artifact requirements. The generator is instructed that the task must remain meaningful after omission and that the excluded dimension should have a genuine communicative role rather than being an arbitrary token constraint.

Candidate-generation prompt template. The construction prompt asks the generator to produce diverse realistic scenarios across the six target domains and to return structured records with the following fields: `domain`, `scenario`, `writing_instruction`, `excluded_dimension`, `retained_content`, `difficulty`, and `artifact_requirements`. It further requires (i) semantic rather than lexical exclusion, (ii) feasibility after removing the target dimension, (iii) diversity in exclusion type and difficulty, and (iv) no near-duplicate or trivial prompts. Candidate records are parsed deterministically and passed through the validation and curation steps described in Appendix A.1.

```
Generate realistic writing scenarios for evaluating semantic
omission.  For each scenario, specify the domain, writing
goal, one semantic dimension that can be excluded without
making the task impossible, the content that must remain,
a difficulty level, and any artifact requirements.  The
exclusion must be semantic rather than a banned-word
constraint.  Cover diverse domains, exclusion types, and
difficulty levels; avoid trivial or near-duplicate items.
Return structured records using the required schema.
```

Four task templates. For each retained base scenario, we instantiate four matched prompts while keeping the semantic target and retained content fixed. *Direct Exclusion* supplies source material containing both retained and excluded information and instructs the model to write using the retained information without discussing the target. *Open-ended Scope Exclusion* gives the writing goal and exclusion instruction without explicitly supplying target content. *Rewrite Removal* provides an existing draft containing the target and asks for a revised artifact with that dimension removed while preserving the remaining content. *Multi-turn Update* places the target in earlier dialogue or a previous draft and then introduces a later instruction that narrows the permitted scope. These templates correspond exactly to the four formats evaluated in Section 2.2.

Supplementary release. The frozen benchmark is maintained as `semantic_omission_bench_v1.jsonl`. The supplementary release records the benchmark schema, construction templates, and file hash so that the evaluated set can be matched exactly to the experiment outputs.

A.3 Artifact Evaluation Rubric and Judge Prompt

The evaluator receives the original writing request, the excluded semantic dimension, the retained-content specification, and the complete candidate artifact. For semantic omission, it checks direct mention, semantic paraphrase, polarity reversal, implication, contrast or concession, qualification, meta-acknowledgment, structural anchoring, and other defensible semantic dependence on the excluded dimension.

A leakage judgment must be supported by evidence from the generated artifact; lexical overlap alone is insufficient when the relevant semantics are not invoked. Task fulfillment assesses whether the response follows the requested artifact form, remains topically relevant, contains substantive and coherent content, and sufficiently covers the retained-content requirements. Refusal, vacuous text, or excessive deletion may therefore satisfy pure omission while failing overall validity.

Judge prompt structure. For each candidate artifact, the evaluator receives the original writing request, optional context, the excluded semantic boundary, retained-content requirements, and the complete model output. It returns separate binary judgments for semantic omission and task fulfillment, together with supporting evidence for any leakage judgment. The same fixed prompt structure is used across the diagnostic and repair experiments.

We conduct human evaluation for two complementary purposes. First, we validate whether the fixed rubric-based automatic evaluator reliably captures semantic omission and task fulfillment, particularly in difficult cases involving implicit or rhetorical semantic leakage. Second, we independently verify the central method-level finding that SelectOmit improves instruction-conditional semantic control rather than merely strengthening unconditional suppression; that method comparison is reported in Appendix C.7.

A.4 Human Validation of the Automatic Evaluator

We construct a stratified validation sample of 600 model outputs covering all four task formats, all construction-difficulty levels, multiple model families, and both pass and fail decisions produced by the automatic evaluator. Two annotators familiar with language-model evaluation participate in the study. Each response is independently annotated by two annotators, with a third annotator adjudicating disagreements in the final gold labels. Annotators are blinded to model identity, training method, and automatic evaluator decisions. For each item, annotators see the original writing request, the excluded semantic boundary, the retained-content specification, and the complete candidate artifact. They answer four questions:

1. **Semantic omission:** Does the final artifact meaningfully invoke the excluded semantic dimension?
2. **Leakage type:** If omission fails, which forms of leakage are present?
3. **Task fulfillment:** Does the response satisfy the requested writing task and retained-content requirements?
4. **Overall validity:** Does the response satisfy both the semantic requirement and the writing task?

Inter-annotator agreement is high despite the semantic nature of the task. Cohen's κ is 0.82 for semantic omission and 0.85 for task fulfillment, with raw pairwise agreement of 90.8% and 92.5%, respectively.

Using the adjudicated human labels as reference, the fixed automatic evaluator achieves high agreement with human judgment.

Evaluator errors are concentrated in the most semantically diffuse cases. False positives most often arise when an output discusses a neighboring concept that is strongly associated with, but does

Table 7: Agreement of the fixed rubric-based evaluator with adjudicated human labels on the 600-response validation sample. Precision, recall, and F1 treat satisfaction of the corresponding criterion as the positive class.

Criterion	Precision	Recall	F1
Semantic omission	0.93	0.91	0.92
Task fulfillment	0.96	0.94	0.95
Overall validity	0.92	0.90	0.91

not substantively invoke, the excluded dimension. False negatives are more common when target semantics are distributed across multiple sentences through implication or discourse-level framing rather than expressed in a localized phrase. Agreement is considerably higher for direct mentions, explicit paraphrases, polarity reversals, and clear contrastive leakage.

B Diagnostic Details and Additional Results

B.1 Boundary Identification Probe

For every benchmark item, the boundary-identification probe asks the model to explicitly state the semantic dimension that should be excluded and the content that should remain. To match the analysis in Section 3.2, we use the first probe response for each benchmark item to determine whether the boundary is correctly identified, and then evaluate writing leakage only on the corresponding identified items. This probe is treated as a behavioral measure of explicit articulation.

Table 8: Explicit boundary identification and subsequent semantic leakage. Identification is computed from the first probe response for each of the 400 benchmark items. Leakage is measured over writing responses associated with correctly identified items.

Model	Identified Items	Identification (%)	Leakage after Identification (%)
Qwen3-4B	315 / 400	78.75	17.99
Qwen3-8B	287 / 400	71.75	11.73
Llama-3.1-8B	243 / 400	60.75	7.68
GLM-4-9B	274 / 400	68.50	12.41
Gemma-2-9B	288 / 400	72.00	3.36

The probe scoring focuses on explicit semantic identification of the exclusion boundary rather than strict serialization or JSON-format adherence. Formatting failures are therefore recorded separately and are not treated as failures of boundary identification.

B.2 Boundary-Clarification Interventions

Section 3.3 evaluates four interventions that progressively modify how the semantic boundary is presented. *Strong* explicitly extends the prohibition to indirect and rhetorical references; *Self* requires the model to restate the boundary before writing; *Oracle* directly supplies the reference boundary and retained focus; and *Focus-only* replaces the negative constraint with positive retained-content guidance.

For completeness, the Strong intervention is also evaluated across all five diagnostic checkpoints.

B.3 Diagnosis on Larger-Scale Models

The main diagnostic study focuses on open-weight models in the 4B–9B range because these checkpoints can subsequently be adapted in the repair experiments. To examine whether semantic omission remains challenging for stronger API-accessible models, we conduct a complementary diagnosis on GPT-5.6 Sol, Gemini 3.1 Pro Preview, GLM-5.3, Qwen3.8-Max, and DeepSeek-V4.1-Flash. Our goal is to test whether stronger models naturally eliminate the semantic-omission behavior observed in the smaller trainable checkpoints.

Table 9: Valid success (%) under the stronger exclusion instruction.

Model	Baseline	Strong
Qwen3-4B	80.42	85.17
Qwen3-8B	86.67	89.67
Llama-3.1-8B	91.08	93.33
GLM-4-9B	83.25	90.58
Gemma-2-9B	95.25	96.25

The same fixed rubric evaluator is used for comparability with the primary benchmark. Because the evaluator itself belongs to the GPT family, outputs from GPT-5.6 Sol are deliberately included in the human evaluator-validation sample reported in Appendix A.4.

B.3.1 RQ1: Does Semantic Omission Failure Persist in Larger Models?

We first repeat the four-format benchmark evaluation to determine whether semantic omission failures persist in stronger models and whether the task-dependent pattern observed in the main study remains visible.

Table 10: Valid success (%) of stronger API-accessible models across the four semantic omission task formats.

Model	Direct	Open	Rewrite	Multi-turn	Overall
GPT-5.6 Sol	92.67	94.33	81.00	97.67	91.42
Gemini 3.1 Pro	94.33	96.00	86.67	97.33	93.58
GLM-5.3	85.33	89.67	70.67	94.00	84.92
Qwen3.8-Max	89.67	92.00	77.67	96.00	88.83
DeepSeek-V4.1-Flash	88.33	91.33	75.67	95.33	87.67

The results show that semantic omission remains non-trivial even for substantially stronger models. Overall valid success ranges from 84.92% to 93.58%, and the API-accessible models do not uniformly outperform the smaller open-weight checkpoints in the main evaluation. In particular, performance varies substantially across model families, suggesting that semantic omission does not exhibit a simple monotonic relationship with general model capability or scale.

The task-dependent pattern observed in the main diagnosis also persists. Multi-turn Update is relatively robust across models, whereas Rewrite Removal remains substantially more difficult, with valid success ranging from 70.67% to 86.67%. This gap is especially notable because rewrite tasks explicitly expose the semantic content that must subsequently be removed, making the excluded dimension highly salient during generation.

These results suggest that stronger general-purpose models do not automatically eliminate semantic omission failures. Some models substantially improve over the smaller checkpoints, whereas others remain in a similar performance regime. Semantic omission therefore appears to depend on model-specific generation behavior rather than following a simple scaling trend.

B.3.2 RQ2: Does Explicit Boundary Identification Guarantee Omission in Larger Models?

We next apply the same boundary-identification probe used in Section 3.2. For each model, we measure how often the exclusion boundary is correctly identified and, among those items, how often the excluded semantic dimension nevertheless reappears during subsequent writing.

Explicit boundary identification is generally stronger than for the smaller checkpoints, but the improvement is neither uniform nor sufficient to guarantee successful omission. Explicit boundary-identification rates range from 76.50% to 91.25%, showing substantial variation across model families.

More importantly, semantic leakage remains non-negligible even after restricting the analysis to correctly identified items. Leakage after identification ranges from 5.66% to 12.31%. Thus, models

Table 11: Explicit boundary identification and subsequent semantic leakage for stronger API-accessible models. Leakage is measured only over writing responses associated with benchmark items whose exclusion boundary was correctly identified by the corresponding model.

Model	Identified Items	Identification (%)	Leakage after Identification (%)
GPT-5.6 Sol	356 / 400	89.00	7.40
Gemini 3.1 Pro	365 / 400	91.25	5.66
GLM-5.3	306 / 400	76.50	12.31
Qwen3.8-Max	329 / 400	82.25	9.93
DeepSeek-V4.1-Flash	320 / 400	80.00	10.83

that can explicitly identify what should be excluded may nevertheless reintroduce the same semantic dimension while constructing the final artifact.

The larger-model results therefore reproduce the central behavioral pattern observed in Section 3.2. Better explicit boundary identification is associated with stronger omission performance, but the two are not equivalent. These findings are consistent with a gap between explicit boundary identification and reliable artifact-level execution rather than with semantic omission being reducible to negative-instruction misunderstanding alone.

B.3.3 RQ3: Can Explicit Boundary Information Solve the Problem?

Finally, we repeat the Strong, Self, Oracle, and Focus-only interventions from Section 3.3. This tests whether stronger models can reliably translate increasingly explicit semantic-boundary information into artifact-level omission.

Table 12: Valid success (%) of stronger API-accessible models under increasingly explicit semantic-boundary interventions.

Model	Baseline	Strong	Self	Oracle	Focus-only
GPT-5.6 Sol	91.42	93.08	94.00	95.25	93.83
Gemini 3.1 Pro	93.58	94.50	95.08	96.17	95.33
GLM-5.3	84.92	87.75	89.08	91.42	90.17
Qwen3.8-Max	88.83	90.42	91.58	93.17	92.25
DeepSeek-V4.1-Flash	87.67	89.33	90.50	92.08	91.00

Making the exclusion boundary more explicit consistently improves semantic omission across models. Stronger prohibitions provide an initial improvement, asking the model to restate the boundary further improves performance, and directly supplying the reference boundary under Oracle produces the strongest result for every evaluated model.

However, explicit boundary information does not eliminate semantic omission failures. Even under Oracle, valid success ranges from 91.42% to 96.17%, leaving residual error rates between 3.83% and 8.58%. The persistence of these failures is particularly informative because the intended exclusion boundary is directly supplied rather than inferred by the model.

Taken together, the larger-model diagnosis yields two observations. First, semantic omission does not exhibit a simple scaling trend: stronger API-accessible models vary substantially, and several remain in a performance regime comparable to much smaller checkpoints. Second, across model families, correctly identifying or explicitly receiving the semantic boundary still does not guarantee that the boundary will be maintained throughout generation. These findings reinforce the motivation for treating semantic omission as a problem of reliable conditional semantic control rather than assuming that it will disappear automatically with increasing model capability.

B.4 Inference and Evaluator Scope

Diagnostic experiments use stochastic decoding with three responses per item, whereas repair experiments use one greedy response per prompt. Consequently, the Base-model values reported in the two settings need not be numerically identical.

All evaluated writing uses a maximum completion length of 384 tokens. The fixed automated evaluator is logged as `gpt-5.6-sol`. All judgments, supporting evidence, and rubric versions are retained for audit and human validation.

C Repair Details and Additional Experiments

C.1 SelectOmit Training Data

The SelectOmit training data are constructed to support the three objectives introduced in Section 4.1: Targeted Suppression, Artifact Preservation, and Conditionality. For Targeted Suppression, each training group contains an exclusion prompt together with a leaking response and localized annotations of the response spans that invoke the excluded semantic dimension. For Artifact Preservation, the same exclusion setting additionally contains a clean response that successfully completes the task and a utility-negative response that satisfies omission but produces an inadequate writing artifact. For Conditionality, a subset of groups further contains a counterfactual inclusion instruction and response for the same semantic target.

The resulting dataset contains 946 training groups drawn from 313 scenario identifiers. All 946 groups contain a leaking response, a clean response, and a utility-negative response, while 145 groups additionally contain counterfactual inclusion supervision. Across task formats, the groups contain 194 Direct Exclusion, 200 Open-ended Scope Exclusion, 359 Rewrite Removal, and 193 Multi-turn Update examples. Across domains, the training groups contain 155 academic-writing, 165 product/marketing, 160 professional-writing, 157 reports/analysis, 152 public-communication, and 157 reviews/recommendations examples.

Table 13: Summary of the SelectOmit training data.

Training-data component	Count
Training groups	946
Distinct scenario identifiers	313
Leaking responses	946
Clean responses	946
Utility-negative responses	946
Counterfactual inclusion groups	145
Merged suppression spans	953
Positive inclusion spans	195

Span annotations are stored as character intervals over raw assistant responses and mapped to overlapping completion tokens during training. Unannotated tokens in a leaking response are not automatically treated as positive targets; the separate clean response provides the complete positive writing target.

C.2 Utility-Negative Construction

Utility negatives are designed to satisfy semantic omission while failing to produce an adequate writing artifact. This allows the utility-preference objective to distinguish useful omission from degenerate content avoidance.

Table 14: Distribution of utility-negative failure types.

Utility failure type	Count
Over-broad suppression	172
Overly brief response	169
Partial refusal	169
Generic response	153
Meta-acknowledgment instead of artifact	146
Insufficient retained-content coverage	137

A meta-acknowledgment belongs to this set only when it does not itself invoke the excluded semantic dimension. If the acknowledgment explicitly reintroduces that dimension, it is treated as semantic leakage instead.

C.3 Data Auditing and Split Integrity

A GLM-32B cross-audit of 200 machine-filtered training pairs produced 199 valid judgments and one parse failure. The auditor agreed with all 199 clean-response labels and with 173 of the 199 negative-response labels (86.93%). Agreement is lower for several small rhetorical subcategories, motivating the human validation described in Appendix A.4.

The 400-item held-out benchmark has no recorded overlap with the training data in scenario identifiers, pair-group identifiers, or exact excluded-dimension strings. These checks prevent direct instance overlap.

The instruction-switching suite pairs each held-out exclusion task with an inclusion instruction for the same semantic target. The compositional suite uses 100 scenarios with two semantic dimensions and constructs the four possible include/exclude combinations. The conditional-likelihood analysis uses 300 source-exposed items from the Direct Exclusion, Rewrite Removal, and Multi-turn Update formats and pairs each target span with a nearby non-target span of similar length.

C.4 SelectOmit Training and Implementation

SelectOmit performs single-stage mixed-supervision post-training. Training examples supporting Targeted Suppression, Artifact Preservation, and Conditionality are jointly sampled, and each example activates the loss terms for which supervision is available. The model starts from the original instruction-tuned checkpoint with newly initialized LoRA parameters (Hu et al., 2022). The unadapted checkpoint is used as the fixed reference model for the utility-preference objective.

The trainer uses teacher-forced response probabilities and does not call the external evaluator during optimization. We use AdamW (Loshchilov & Hutter, 2019) with $(0.9, 0.999)$ moments, numerical epsilon 10^{-8}, weight decay 0.01, cosine learning-rate decay, gradient clipping at 1.0, gradient checkpointing, and BF16 base-model loading.

Table 15: SelectOmit training configuration.

SelectOmit setting	**Value**
Training groups / inclusion counterparts	946 / 145
Epochs	2
Learning rate	5×10^{-5}
LoRA rank / alpha / dropout	32 / 64 / 0.05
Microbatch groups / accumulation	1 / 2
Maximum sequence length	2,048
Utility-reference batch size	4
DPO temperature β	0.1
$(\lambda_o, \lambda_c, \lambda_u, \lambda_+)$	$(1, 1, 0.5, 0.5)$
Training seed	20260827
Method evaluation	Greedy, 384 new tokens

In the single-group microbatch configuration, Equation 6 uses mean unlikelihood over annotated leakage tokens, mean clean-response CE over completion tokens, response-summed log probabilities for the utility-preference objective, and mean inclusion CE over target tokens. The inclusion contribution is zero for groups without a counterfactual inclusion counterpart.

C.4.1 Baseline Implementations

Matched SFT uses the same 946 source groups and trains on the corresponding clean responses. Mixed SFT uses 946 clean responses plus 237 auxiliary boundary-identification examples. Standard DPO starts from the Mixed-SFT adapter and trains for one additional epoch on 946 clean-versus-leaking preference pairs, using a learning rate of 5×10^{-6} and $\beta = 0.1$.

Positive-matched SFT receives the same 946 clean responses and 145 counterfactual inclusion responses available to SELECTOMIT. It applies completion-only CE to the full inclusion responses, whereas SELECTOMIT applies localized inclusion supervision to the annotated target spans. Span-matched Conditional SFT uses the same matched exclusion/inclusion data but applies inclusion cross-entropy only on the annotated target spans; it does not use localized unlikelihood or the utility-preference objective.

The response-wide unlikelihood ablation retains the leaking response but applies negative supervision to the full response rather than the annotated leakage spans. The no-clean, no-utility, and no-inclusion ablations remove the corresponding loss terms from Equation 6.

C.5 INSTRUCTION-SWITCHING AND COMPOSITIONAL EVALUATION

Under an inclusion instruction, the semantic criterion is reversed: the designated semantic dimension must be substantively and naturally covered in the final artifact. In the two-dimension compositional evaluation, each dimension is judged independently according to its requested include/exclude mode, after which task fulfillment is evaluated.

Four-condition success requires four separately generated artifacts for the same scenario—include both, exclude A/include B, include A/exclude B, and exclude both—to all satisfy their corresponding semantic and task requirements.

C.6 CONDITIONAL LIKELIHOOD ANALYSIS

To complement generation-level switching and compositional evaluation, we examine whether instruction-dependent control is also reflected in the model's conditional probabilities. For a target span s, we define the instruction-induced likelihood change as

$$G(s) = \ell_\theta(s \mid x^+) - \ell_\theta(s \mid x^-),$$

and compare it against a length-matched non-target control span:

$$Q = G(s_{\text{target}}) - G(s_{\text{control}}).$$

Positive Q indicates that switching from exclusion to inclusion preferentially restores the target semantic content relative to ordinary surrounding content. As shown in Figure 5, the selectivity scores for Base, Matched SFT, the model without inclusion supervision, and full SELECTOMIT are -0.320, 0.156, 4.025, and 5.014, respectively. Inclusion supervision therefore primarily increases target availability under inclusion instructions while preserving strong suppression under exclusion instructions.

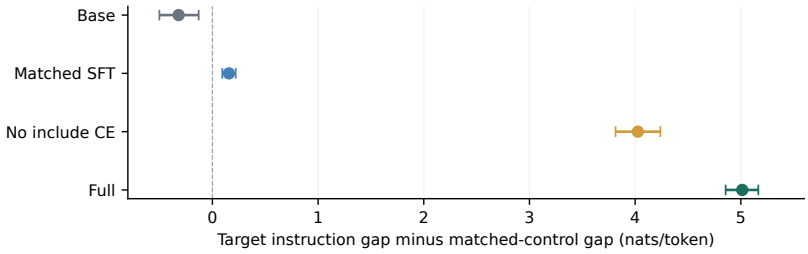

Figure 5: Conditional selectivity Q in nats per token. Positive values indicate that switching from exclusion to inclusion preferentially restores the target semantic content relative to matched non-target content.

C.7 HUMAN METHOD COMPARISON

We next conduct a second blinded study to independently evaluate the central repair claim. We evaluate Base, Positive-matched SFT, Standard DPO, and SELECTOMIT on all 100 held-out scenarios used in the instruction-switching evaluation. Each method generates one response under the

exclusion instruction and one response under the corresponding inclusion instruction, yielding 800 candidate artifacts in total.

The same annotation criteria are used, and annotators are blinded to system identity and training method. A scenario is counted as paired-valid only when the same method successfully handles both the exclusion and inclusion instruction for the corresponding semantic target.

Table 16: Human evaluation of instruction switching on 100 held-out scenarios. Paired validity requires successful completion of both the exclusion and inclusion condition for the same semantic target.

Method	**Exclusion Valid (%)**	**Inclusion Valid (%)**	**Paired Valid (%)**
Base	81.0	92.0	74.0
Positive-matched SFT	93.0	77.0	72.0
Standard DPO	96.0	65.0	63.0
SELECTOMIT	97.0	84.0	82.0

The human judgments reproduce the main pattern observed with automatic evaluation. Standard DPO is highly effective under exclusion instructions, increasing exclusion validity to 96%, but its inclusion validity falls to 65%. As a result, only 63% of scenarios are successfully completed under both instructions.

In contrast, SELECTOMIT retains similarly strong exclusion performance at 97% while achieving substantially higher inclusion validity at 84%. Its paired validity therefore reaches 82%, compared with 74% for the Base model, 72% for Positive-matched SFT, and 63% for Standard DPO. This supports the interpretation that SELECTOMIT improves instruction-conditioned semantic control rather than simply learning a stronger general tendency to avoid the target semantics.

C.8 HUMAN-EVALUATION SUMMARY

Table 17: Summary of the human-evaluation study.

Item	**Human-evaluation result**
Evaluator-validation sample	600 responses stratified across task format, difficulty, model family, evaluator decision, and leakage type
Annotators	Four annotators; two independent annotations per response with third-annotator adjudication for disagreements
Human–human agreement	$\kappa = 0.82$ for semantic omission and $\kappa = 0.85$ for task fulfillment
Judge–human agreement	Semantic-omission F1 = 0.92, task-fulfillment F1 = 0.95, and overall-validity F1 = 0.91
Method-comparison sample	100 held-out scenarios, four systems, and matched exclusion/inclusion instructions, yielding 800 outputs
Human exclusion validity	Base 81% / Positive SFT 93% / DPO 96% / SELECTOMIT 97%
Human inclusion validity	Base 92% / Positive SFT 77% / DPO 65% / SELECTOMIT 84%
Human paired switching	Base 74% / Positive SFT 72% / DPO 63% / SELECTOMIT 82%

C.9 SUPPLEMENTARY EVALUATION ON AUTHENTIC SOURCE MATERIALS

This experiment examines whether the same behavior appears when the writing context is drawn from authentic published source material rather than synthetic benchmark scenarios. We construct exclusion/inclusion pairs around 50 documents: 15 Mozilla Support pages, 15 GOV.UK releases, and 20 PLOS ONE articles. Half of the scenarios are writing tasks and half are rewriting tasks. Each document provides a source excerpt, three retained-content requirements, and one semantic target whose instruction is switched between exclusion and inclusion.

C.9.1 MATERIALS AND CONSTRUCTION

The 50 documents were collected from three public sources: Mozilla Support, GOV.UK, and PLOS ONE. Source records preserve titles, URLs, attribution, license evidence, excerpt selections, and hashes. The recorded licenses are Mozilla's CC BY-SA 3.0 or later, the UK Open Government Licence for the selected GOV.UK text, and article-specific CC BY terms for PLOS. All annotated target spans and retained-content evidence spans were mechanically checked against their source excerpts.

C.9.2 GENERATION AND EVALUATION

We reuse the existing Qwen3-4B checkpoints with greedy decoding and no additional training. The semantic criterion is identical to that used in the main evaluation. Task fulfillment is assessed by artifact compliance, topical relevance, substantive coherence, and retained-content coverage. The supplementary task score does not include the original word-count bound.

C.9.3 RESULTS

Table 18: Authentic-source writing and rewriting results (%) on 50 documents. Each document provides a matched exclusion/inclusion pair.

Model	V^-	V^+	V_{pair}
Base	60.0	100.0	60.0
Matched SFT	60.0	96.0	58.0
Positive-matched SFT	72.0	96.0	70.0
Standard DPO	78.0	68.0	58.0
SELECTOMIT	**84.0**	94.0	**80.0**

Across the 50 documents, SELECTOMIT reaches 84% exclusion validity, 94% inclusion validity, and 80% paired validity. Standard DPO remains competitive under exclusion (78%), but its inclusion validity drops to 68%, resulting in only 58% paired validity. Thus, the main difference is not simply stronger suppression: SELECTOMIT is substantially more reliable when the same semantic content must become available again under the opposite instruction. These results provide a supplementary check that the conditional-control behavior observed on the synthetic benchmark transfers to source-grounded writing and rewriting tasks.

Table 19: Content-based task fulfillment (%) on the authentic-source evaluation.

Model	T^-	T^+
Base	100.0	100.0
Matched SFT	100.0	100.0
Positive-matched SFT	100.0	100.0
Standard DPO	100.0	98.0
SELECTOMIT	98.0	96.0

Task fulfillment remains high across this evaluation: 496 of the 500 responses pass the content-based task criterion.

C.9.4 PAIRED STATISTICAL TESTS

The sampling unit is a document with both instruction conditions. We report exact two-sided McNemar tests comparing SELECTOMIT with the four comparison methods, with Holm adjustment separately within exclusion validity and paired validity.

C.10 ADDITIONAL DEVELOPMENT CONTROLS

The source-control experiment compares 60 paired prompts in which the target semantic content is either exposed or sanitized in the source. These controls complement the main diagnosis but have limited discriminative power once adapted models approach ceiling performance.

Table 20: Exact McNemar comparisons between SELECTOMIT and comparison methods on the 50 authentic-source documents.

Comparison	**Δ (pp)**	**Exact p**	**Holm p**
Exclusion validity			
vs Base	+24.0	.0018	.0073
vs Matched SFT	+24.0	.0018	.0073
vs Positive-matched SFT	+12.0	.1094	.2188
vs DPO	+6.0	.4531	.4531
Paired validity			
vs Base	+20.0	.0063	.0137
vs Matched SFT	+22.0	.0034	.0137
vs Positive-matched SFT	+10.0	.1797	.1797
vs DPO	**+22.0**	**.0034**	**.0137**

Table 21: Development and source-control valid success (%).

Model	**Original Dev**	**Exposed**	**Sanitized**
Base	84.00	78.33	96.67
Matched SFT	100.00	100.00	100.00
Mixed SFT	100.00	100.00	100.00
Standard DPO	99.50	100.00	100.00
Positive-matched SFT	100.00	98.33	98.33
SELECTOMIT	100.00	100.00	100.00

C.11 Statistical Analysis and Uncertainty

For the primary held-out benchmark and derived conditional-control evaluations, we use 20,000 paired percentile-bootstrap draws over scenario clusters with seed 20260915. All four task variants for a scenario remain grouped during held-out exclusion resampling, while both inclusion and exclusion conditions remain paired in the instruction-switching evaluation. Compositional all-four success is computed at the scenario level before resampling. The conditional-likelihood analysis uses the same scenario-cluster bootstrap procedure.

The reported intervals quantify variation across evaluation scenarios conditional on the trained checkpoints. They do not capture training-seed variability or uncertainty in the automated evaluator. Unless otherwise specified, intervals are unadjusted 95% percentile intervals.

Table 22: Held-out exclusion contrasts using paired scenario-cluster bootstrap intervals.

Comparison	ΔV **(pp)**	**95% cluster CI**
SELECTOMIT – Base	+17.50	$[+13.75, +21.25]$
SELECTOMIT – Matched SFT	+4.25	$[+2.00, +6.75]$
SELECTOMIT – Mixed SFT	+4.00	$[+1.50, +6.50]$
SELECTOMIT – Standard DPO	+1.00	$[-1.25, +3.25]$
SELECTOMIT – Positive-matched SFT	+4.25	$[+2.00, +6.50]$
SELECTOMIT – w/o Span UL	+4.25	$[+2.25, +6.50]$
SELECTOMIT – Response-wide UL	+3.00	$[+0.75, +5.50]$
SELECTOMIT – w/o Clean CE	+64.00	$[+59.25, +68.75]$
SELECTOMIT – w/o Utility Preference	+1.25	$[-1.00, +3.75]$
SELECTOMIT – w/o Inclusion Supervision	+1.00	$[-0.75, +2.75]$

Table 23: SELECTOMIT improvements on instruction switching and two-dimension compositional control.

Comparison	**Outcome**	Δ **(pp)**	**95% cluster CI**
Base	Paired switch	+7.50	$[+2.50, +12.50]$
Matched SFT	Paired switch	+8.25	$[+3.50, +13.00]$
Standard DPO	Paired switch	+18.75	$[+14.00, +23.50]$
Positive-matched SFT	Paired switch	+10.25	$[+5.75, +14.75]$
w/o Inclusion	Paired switch	+10.25	$[+5.50, +15.00]$
Base	All-four	+16.00	$[+3.00, +29.00]$
Matched SFT	All-four	+15.00	$[+4.00, +26.00]$
Standard DPO	All-four	+37.00	$[+25.00, +48.00]$
Positive-matched SFT	All-four	+22.00	$[+11.00, +33.00]$
w/o Inclusion	All-four	+7.00	$[-3.00, +17.00]$