

Supplement for: The Emergent Symbolic Structure of Artificial Neural Networks

R. Thomas McCoy

Paul Soulos

Tal Linzen

Paul Smolensky

S1 Context-free grammars for experiments with GPT-OSS performing syntactic transformations

The context-free grammars (CFGs) that were used for passivization inputs, tense reinflection inputs, and question formation inputs are in Figures S1.1, S1.2, and S1.3, respectively. We do not intend for these CFG to capture reasonable syntactic analyses of the English sentences in question; rather, their purpose is simply to serve as a concise way to express somewhat variable templates for the stimuli that we generate.

S2 Additional details for causal interventions on GPT-OSS

Carrying out causal interventions that change a token’s role carries certain complications that do not apply when only changing a filler; see examples in Tables 2 and 3. Specifically, if a token appears before both the original position and new position of the edited token, it receives no edit; if it is after both the original and new position, then it swaps out one role-filler pair for another, where the two are the same as each other except for their role; if it is after the new position but not the current position, then it has a new role-filler pair added in but nothing subtracted; if it is after the current position but not the new position, then it has one role-filler pair subtracted without adding anything in.

Tables 4 and 5 provide detailed metrics across all causal interventions that we run in the settings of GPT-OSS in symbolic domains (see the tables at the end of this subsection for descriptions and examples of all intervention types). Each table shows 5 metrics; in the following definitions of these metrics, the word *edit* refers to a single addition or subtraction of a role-filler pair (e.g., in *clever* row of Table 2, there are 7 edits):

- **Target acc.:** The accuracy that the target model (GPT-OSS) achieves on the prompts that should result from performing our intervention. That is, our interventions attempt to turn the representation from input 1 into a representation for input 2; this metric shows how well GPT-OSS performs when it is given its actual representation input 2, as this number is likely to be an upper bound on how accurate the interventions can be.
- **Intervention (all):** The standard intervention performance—specifically, the performance on interventions for which we make all changes that are predicted by our hypothesis to be necessary to convert the representation of input 1 into the representation of input 2. For instance, in the examples in Tables 1, 2, and 3, the *intervention (all)* version of the intervention would entail making all edits shown in the “Edit to make” column.
- **Intervention (other):** An ablated version of intervention performance—specifically, the performance on interventions for which we make all changes used in *intervention (other)* except for the changes that involve the token whose representation is being altered. For instance, in Table 1, we would make all edits in the “Edit to make” column except for the edits in the *spy* row; and in Tables 2 and 3, we would make all edits in the “Edit to make” column except for the edits in the *clever* row. If multiple tokens are targeted by the intervention, the edits that are retained are those for which the filler within the added or subtracted role-filler pair is not the one present in the input token whose representation is being edited. E.g., in an intervention that moves the entire prepositional phrase *near the intern*, we would retain the edits to the representation of *the* that involve *near* or *intern* but not the edits that involve this instance of *the* (we say “this instance of” because we would retain edits that involve instances of *the* other than the one inside the prepositional phrase being moved).

$S \rightarrow NP_RC\ V\ NP_RC\ Punct$
$NP_RC \rightarrow Det\ NP$
$NP_RC \rightarrow Det\ NP\ RC$
$NP \rightarrow NP_small$
$NP \rightarrow NP_small\ P\ Det\ NP_small$
$NP_small \rightarrow N$
$NP_small \rightarrow Adj\ N$
$RC \rightarrow Comp\ Det\ NP\ V$
$RC \rightarrow Comp\ V\ Det\ NP$
$Punct \rightarrow .$
$Det \rightarrow the$
$Comp \rightarrow who$
$N \rightarrow \{senator \mid journalist \mid composer \mid director \mid \dots \mid athlete\}$
$V \rightarrow \{recommended \mid visited \mid helped \mid supported \mid \dots \mid admired\}$
$Adj \rightarrow \{happy \mid famous \mid clever \mid tall \mid \dots \mid polite\}$
$P \rightarrow \{near \mid next\ to \mid in\ front\ of \mid behind \mid across\ from\}$

Figure S1.1: Context-free grammar for the passivization task. Some of the lists of terminals have been truncated for space; overall there are 68 nouns (*N*), 15 verbs (*V*), and 10 adjectives (*Adj*). See the project GitHub for the complete word lists.

	Original	Edit to make
The	{subj_det-subj_det:The}	None
spy	{subj_det-subj_n:The, subj_n-subj_n:spy}	– subj_n-subj_n:spy + subj_n-subj_n:astronaut
helped	{subj_det-verb:The, subj_n-verb:spy, verb-verb:helped}	– subj_n-verb:spy + subj_n-verb:astronaut
the	{subj_det-obj_det:The, subj_n-obj_det:spy, verb- obj_det:helped, obj_det- obj_det:the}	– subj_n-obj_det:spy + subj_n-obj_det:astronaut
poet	{subj_det-obj_n:The, subj_n-obj_n:spy, verb-obj_n:helped, obj_det-obj_n:the, obj_n- obj_n:poet}	– subj_n-obj_n:spy + subj_n-obj_n:astronaut
.	{subj_det-punct:The, subj_n-punct:spy, verb-punct:helped, obj_det- punct:the, obj_n-punct:poet, punct-punct:..}	– subj_n-punct:spy + subj_n-punct:astronaut

Table 1: Representational edits that would be made with the goal of changing *spy* to *astronaut*.

```

S → NP_RC_sg Vsg NP_RC Punct
S → NP_RC_pl Vpl NP_RC Punct

NP_RC_sg → Det NP_sg
NP_RC_sg → Det NP_sg RC_sg
NP_RC_pl → Det NP_pl
NP_RC_pl → Det NP_pl RC_pl

NP_sg → NP_sg_small
NP_sg → NP_sg_small P Det NP_sg_small
NP_pl → NP_pl_small
NP_pl → NP_pl_small P Det NP_pl_small

NP_sg_small → Nsg
NP_sg_small → Adj Nsg
NP_pl_small → Npl
NP_pl_small → Adj Npl

RC_sg → Comp Det NP_sg Vsg
RC_sg → Comp Det NP_pl Vpl
RC_sg → Comp Vsg Det NP
RC_pl → Comp Det NP_sg Vsg
RC_pl → Comp Det NP_pl Vpl
RC_pl → Comp Vpl Det NP

NP → NP_sg
NP → NP_pl

NP_RC → NP_RC_sg
NP_RC → NP_RC_pl

Punct → .
Det → the
Comp → who
Nsg → {senator | journalist | composer | director | ... | athlete }
Npl → {senators | journalists | composers | directors | ... | athletes }
Vsg → {recommends | visits | helps | supports | ... | recognizes }
Vpl → {recommend | visit | help | support | ... | recognize }
Adj → {happy | famous | clever | tall | ... | polite }
P → {near | next to | in front of | behind | across from }

```

Figure S1.2: Context-free grammar for the tense reinlection task. Some of the lists of terminals have been truncated for space; overall there are 56 singular nouns (*Nsg*), 56 plural nouns (*Npl*), 14 singular verbs (*Vsg*), 14 plural verbs (*Vpl*), and 10 adjectives (*Adj*). See the project GitHub for the complete word lists.

	Original	Edit to make
The	{subj_det-subj_det:The}	None
spy	{subj_det-subj_n:The, subj_n-subj_n:spy}	+ subj_adj-subj_n:clever
helped	{subj_det-verb:The, subj_n-verb:spy, verb-verb:helped}	+ subj_adj-verb:clever
the	{subj_det-obj_det:The, subj_n-obj_det:spy, verb- obj_det:helped, obj_det- obj_det:the}	+ subj_adj-obj_det:clever
clever	{subj_det-obj_adj:The, subj_n-obj_adj:spy, verb-obj_adj:helped, obj_det-obj_adj:the, obj_adj-obj_adj:clever}	- subj_det-obj_adj:The + subj_det-subj_adj:The - obj_adj-obj_adj:clever + subj_adj-subj_adj:clever - subj_n-obj_adj:spy - verb-obj_adj:helped - obj_det-obj_adj:the
poet	{subj_det-obj_n:The, subj_n-obj_n:spy, verb-obj_n:helped, obj_det- obj_n:the, obj_adj- obj_n:clever, obj_n- obj_n:poet}	- obj_adj-obj_n:clever + subj_adj-obj_n:clever
.	{subj_det-punct:The, subj_n-punct:spy, verb-punct:helped, obj_det- punct:the, obj_adj- punct:clever, obj_n-punct:poet, punct-punct:.}	- obj_adj-punct:clever + subj_adj-punct:clever

Table 2: Representational edits that would be made with the goal of moving *clever* from the object to the subject.

```

S → NP_RC Aux V NP_RC Punct

NP_RC → Det NP
NP_RC → Det NP RC

NP → NP_small
NP → NP_small P Det NP_small

NP_small → N
NP_small → Adj N

RC → Comp Det NP Aux V
RC → Comp Aux V Det NP

Punct → .
Det → the
Comp → who
N → {senator | journalist | composer | director | ... | athlete }
V → {recommend | visit | help | support | ... | admire }
Adj → {happy | famous | clever | tall | ... | polite }
P → {near | next to | in front of | behind | across from }
Aux → {can | will | would | could | should }

```

Figure S1.3: Context-free grammar for the question formation task. Some of the lists of terminals have been truncated for space; overall there are 68 nouns (*N*), 15 verbs (*V*), and 10 adjectives (*Adj*). See the project GitHub for the complete word lists.

- **Intervention (self):** Another ablated version of intervention performance. This metric is the opposite of *intervention (other)*: it entails making only the edits that target the token whose representation is being edited. For instance, in Table 1, we would only make the edits that are in the *spy* row, and in Table 2, we would only make the edits that are in the *clever* row.
- **Intervention (replace):** This condition is a modified version of *intervention (self)*. However, instead of partially modifying a representation, in this condition we entirely overwrite it. The specific version that this takes is that the representations that would be edited under *intervention (self)* are changed to the sum of the *intervention (self)* edits that have a plus sign. E.g., in Table 1, the representation at the position of *spy* would be changed to *subj_n-subj_n:astronaut*, and in Table 2, the representation of *clever* would be changed to *subj_det-subj_adj:The + subj_adj-subj_adj:clever*.

These 4 intervention metrics are reported in two ways: as accuracies (Table 4), which show the proportion of tests on which GPT-OSS produces the correct response when fed the edited representations, and as agreements (Table 5), which show the proportion of tests on which GPT-OSS produces the same response when fed the edited representations as it would have produced if it were given the revised prompt that the edits are intended to yield. In both tables, the *target acc.* column uses accuracies (i.e., it is the same across both tables), but the *intervention* columns differ in that Table 4 shows accuracies while Table 5 shows agreements. Unless stated otherwise, all discussion of interventions in the paper uses *intervention (all)* accuracy as the metric for success.

In general, the interventions achieve high accuracies. There are two main exceptions. First, some of the syllogism interventions have fairly low accuracies (Table 4), but their agreements are substantially higher (Table 5), showing that the shortcomings of the interventions most likely derive from errors made by the target model rather than in the intervention process. Second, for passivization, the *r-pp_move* and *r-rc_move* settings have moderately low accuracies. The same categories resulted in high accuracies for tense reinflection and question formation; a possible explanation for the difference across these language tasks is that target model performance and DISCOVER accuracy are both higher in tense reinflection and question formation (approaching 100%) than

for passivization (closer to 90%), which might indicate that passivization is subject to greater uncertainty or noise in the target model and/or the DISCOVER model.

Generally speaking, *intervention (self)* continued to perform almost as well as *intervention (all)* when a filler was being edited, but when a role was being edited (i.e., when we were performing a structural change) performance was substantially in *intervention (self)* than in *intervention (all)*. This result suggests that the representations of token identity are fairly localized to individual tokens in GPT-OSS, while the representations of structure are more distributed across tokens. One counterexample to the trend that filler changes only require local edits is in the case of editing the premises inside syllogisms. For interventions that edit a filler corresponding to a content word such as a noun (e.g., changing *No actors are accountants* to *No lawyers are accountants*), local interventions are reasonably successful (with an accuracy of 0.570). However, for interventions that edit a filler corresponding to a quantifier (e.g., changing *No actors are accountants* to *All actors are accountants*), the local interventions attain very poor performance (with an accuracy of 0.038). One potential explanation is that, in syllogisms, quantifier identity defines the logical structure of the problem while noun identity is extraneous to the logical structure; thus, the pattern that we see with syllogisms is consistent with the broader view that structural information is distributed across tokens.

Below are two tables giving descriptions and examples for all types of interventions that we conduct. The first table shows interventions on fillers, where we change the identity of some filler (indicated by an intervention name starting with *f-*); the second table shows interventions on roles, where we change the position(s) of some filler(s) (indicated by an intervention name starting with *r-*).

Task	Intervention type	Example	Accuracy
Arithmetic	f-number:	Original prompt:	0.978 $\pm$ 0.009
	Change the value of a number	-2 + 3 * -4	
		Intervention (in the “3” token): – {2nd_num:3} + {2nd_num:8}	
		Intended output change: -14 $\rightarrow$ -34	
Syllogisms	f-content: Change a noun to a different noun, or an adjective to a different adjective	Original prompt: No cyclists are nurses. Some cyclists are presidents.	0.836 $\pm$ 0.018
		Intervention (in the “nurses” token): – {sentence1_noun2:nurses} + {sentence1_noun2:golfers}	
		Intended output change: Some presidents are not nurses. $\rightarrow$ Some presidents are not golfers.	
	f-function: Change a quantifier (<i>all</i> , <i>no</i> , or <i>some</i>) to a different one	Original prompt: No clever tutors are famous lawyers. All famous lawyers are golfers.	0.592 $\pm$ 0.026
		Intervention (in the “No” token): – {1st_quantifier:No} + {1st_quantifier:All}	
		Intended output change: Some golfers are not clever tutors. $\rightarrow$ Some golfers are clever tutors.	

Code execution	f-list: Change an element of a list	Original prompt: <pre>def change(sequence): return sequence[:-1] def alter(sequence): return sequence + ["U"] y = ["S", "K", "V"] x = ["I", "V", "R"] change(x)</pre> Intervention (in the “V” token): $-\{x_{2nd:V}\} + \{x_{2nd:N}\}$ Intended output change: $["R", "V", "I"] \rightarrow ["R", "N", "I"]$	0.990 $\pm$ 0.000
	f-affix: Change the letter that is added to a list	Original prompt: <pre>def change(sequence): return ["U"] + sequence def alter(sequence): return sequence + ["A"] x = ["I", "C", "D"] y = ["J", "Z", "Y"] alter(y)</pre> Intervention (in the “A” token): $-\{alter_{9th:A}\} + \{alter_{9th:G}\}$ Intended output change: $["J", "Z", "Y", "A"] \rightarrow ["J", "Z", "Y", "G"]$	1.000 $\pm$ 0.000
	f-rep: Change the number of times a list is repeated	Original prompt: <pre>def change(sequence): return sequence * 2 def alter(sequence): return sequence + ["B"] y = ["J", "R", "M"] x = ["Q", "E", "V"] change(x)</pre> Intervention (in the “2” token): $-\{change_{9th:2}\} + \{change_{9th:3}\}$ Intended output change: $["Q", "E", "V", "Q", "E", "V"] \rightarrow ["Q", "E", "V", "Q", "E", "V", "Q", "E", "V", "Q", "E", "V"]$	1.000 $\pm$ 0.000

	f-arg: Change which variable is the argument of a function call	Original prompt: <pre>def alter(sequence): return sorted(sequence) def change(sequence): return ["H"] + sequence x = ["U", "E", "T"] y = ["X", "Z", "D"] alter(x)</pre>	0.960 $\pm$ 0.020
		Intervention (in the final “x” token): – {arg:x} + {arg:y}	
		Intended output change: ["E", "T", "U"] $\rightarrow$ ["D", "X", "Z"]	
	f-call: Change which function is called	Original prompt: <pre>def change(sequence): return sorted(sequence) def alter(sequence): sequence + ["H"] x = ["Z", "D", "A"] y = ["V", "N", "B"] alter(y)</pre>	0.950 $\pm$ 0.014
		Intervention (in the final “alter” token): – {function_call:alter} + {function_call:change}	
		Intended output change: ["V", "N", "B", "H"] $\rightarrow$ ["B", "N", "V"]	
Passivization	f-content: Change the identity of a noun, verb, or adjective	Original prompt: The coach next to the swimmer helped the illustrator.	0.896 $\pm$ 0.011
		Intervention (in the “illustrator” token): – {object_n:illustrator} + {object_n:spy}	
		Intended output change: The illustrator was helped by the coach next to the swimmer. $\rightarrow$ The spy was helped by the coach next to the swimmer.	

Tense reinflection	f-content: Change the identity of a noun, verb, or adjective	Original prompt: The pilot near the tall astronaut encouraged the cyclist near the manager. Intervention (in the “encouraged” token): – {verb:encouraged} + {verb:visited} Intended output change: The pilot near the tall astronaut encourages the cyclist near the manager. → The pilot near the tall astronaut visits the cyclist near the manager.	0.950 ± 0.000
	f-noun_1change: Change the identity of a noun in a way that also requires 1 verb to change number	Original prompt: The miners avoided the intelligent dancers. Intervention (in the “miners” token): – {subject_n:miners} + {subject_n:technician} Intended output change: The miners avoid the intelligent dancers. → The technician avoids the intelligent dancers.	0.988 ± 0.009
	f-noun_2change: Change the identity of a noun in a way that also requires 2 verbs to change number	Original prompt: The farmer who recommended the famous authors next to the polite philosophers stopped the dancer. Intervention (in the “farmer” token): – {subject_n:farmer} + {subject_n:authors} Intended output change: The farmer who recommends the famous authors next to the polite philosophers stops the dancer. → The authors who recommend the famous authors next to the polite philosophers stop the dancer.	0.990 ± 0.000
Question formation	f-content: Change the identity of a noun, verb, or adjective	Original prompt: The photographer will mention the architect. Intervention (in the “architect” token): – {object_n:architect} + {object_n:miner} Intended output change: Will the photographer mention the architect? → Will the photographer mention the miner?	0.986 ± 0.011
Task	Intervention type	Example	Accuracy

Arithmetic	r-number_swap: Swap the positions of two numbers by changing one's position to the other's, and vice versa	Original prompt: $3 + 6 * 8$ Intervention (in the “3” token): $-\{1st_num:3\} + \{2nd_num:3\}$ Intervention (in the “6” token): $-\{2nd_num:6\} + \{1st_num:6\}$ Intended output change: $51 \rightarrow 30$	0.978 ± 0.017
	r-neg_move: Move a negative sign from one number to another	Original prompt: $8 * 2 + -4$ Intervention (in the “-” token): $-\{neg_3rd:-\} + \{neg_2nd:-\}$ Intended output change: $12 \rightarrow -12$	0.982 ± 0.017
Syllogisms	r-noun_swap: Swap the positions of two nouns by changing one's position to the other's, and vice versa	Original prompt: No actors are accountants. All lawyers are accountants. Intervention (in the “lawyers” token): $-\{sentence2_1st_noun:lawyers\} + \{sentence2_2nd_noun:lawyers\}$ Intervention (in the 2nd “accountants” token): $-\{sentence2_2nd_noun:accountatns\} + \{sentence2_1st_noun:accountants\}$ Intended output change: No lawyers are actors. $\rightarrow$ Some lawyers are not actors.	0.422 ± 0.026 (0.902 ± 0.026)
Code execution	r-list_move: Move an element from the end of one list to the end of the other list	Original prompt: <pre>def change(sequence): return sequence * 4 def alter(sequence): return sequence + ["J"] y = ["G", "P"] x = ["H", "W", "Q"] change(y)</pre> Intervention (in the “Q” token): $-\{x_3rd:Q\} + \{y_3rd:Q\}$ Intended output change: $["G", "P", "G", "P", "G", "P", "G", "P"] \rightarrow ["G", "P", "Q", "G", "P", "Q", "G", "P", "Q", "G", "P", "Q"]$	0.95 ± 0.032

Passivization	r-adj_move: Move an adjective from before one noun to before another	Original prompt: The helpful photographer across from the popular senator hired the coach. Intervention (in the “popular” token): – {subj_pp_adj:popular} + {obj_adj:popular} Intended output change: The coach was hired by the helpful photographer across from the popular senator. → The popular coach was hired by the helpful photographer across from the senator.	0.934 ± 0.018
	r-rc_arg_move: Change a subject relative clause to an object relative clause or vice versa	Original prompt: The intelligent chef who the pilot admired stopped the careful writer. Intervention (in the tokens for “the pilot”; simplified): – {subj_rc_subj:the pilot} + {subj_rc_obj:the pilot} Intended output change: The careful writer was stopped by the intelligent chef who the pilot admired. → The careful writer was stopped by the intelligent chef who admired the pilot.	0.856 ± 0.030
	r-pp_move: Move a prepositional phrase from after one noun to after another	Original prompt: The ambassador across from the technician who recognized the chef supported the clever spy who encouraged the astronaut. Intervention (in the tokens for “across from the technician”; simplified): – {subj_pp:across from the technician} + {subj_rc_obj_pp:across from the technician} Intended output change: The clever spy who encouraged the astronaut was supported by the ambassador across from the technician who recognized the chef. → The clever spy who encouraged the astronaut was supported by the ambassador who recognized the chef across from the technician.	0.696 ± 0.023

r-rc_move: Move a relative clause from after one noun to after another	Original prompt: The careful engineer in front of the nurse who the polite president behind the mayor advised thanked the swimmer.	0.638 $\pm$ 0.017
---	--	-------------------

Intervention (in the tokens for “who the polite president behind the mayor advised”; simplified):

– {subj_rc:who the polite president behind the mayor advised} + {obj_rc:who the polite president behind the mayor advised}

Intended output change:

The swimmer was thanked by the careful engineer in front of the nurse who the polite president behind the mayor advised. → The swimmer who the polite president behind the mayor advised was thanked by the careful engineer in front of the nurse.

Tense reinflection	r-adj_move: Move an adjective from before one noun to before another	Original prompt: The happy journalist avoided the nurse who the ambassadors in front of the politicians mentioned.	0.980 $\pm$ 0.000
--------------------	--	--	-------------------

Intervention (in the “happy” token):

– {subj_adj:happy} + {obj_adj:happy}

Intended output change:

The happy journalist avoids the nurse who the ambassadors in front of the politicians mention. → The journalist avoids the happy nurse who the ambassadors in front of the politicians mention.

r-rc_arg_move_keep: Change a subject relative clause to an object relative clause or vice versa, without needing the verb’s number to change	Original prompt: The writers hired the economists across from the politicians who the polite tourists stopped.	0.986 $\pm$ 0.011
---	--	-------------------

Intervention (in the tokens for “the polite tourists”; simplified):

– {obj_rc_subj:the polite tourists} + {obj_rc_obj:the polite tourists}

Intended output change:

The writers hire the economists across from the politicians who the polite tourists stop. → The writers hire the economists across from the politicians who stop the polite tourists.

r-rc_arg_move_swap Change a subject relative clause to an object relative clause or vice versa, requiring the verb's number to change	Original prompt: The polite tutors helped the explorers who surprised the happy coach. Intervention (in the tokens for “the happy coach”; simplified): – {obj_rc_obj:the happy coach} + {obj_rc_subj:the happy coach}	0.964 ± 0.011
---	--	---------------

Intended output change:
 The polite tutors help the explorers who surprise the happy coach. → The polite tutors help the explorers who the happy coach surprises.

r-pp_move: Move a prepositional phrase from after one noun to after another	Original prompt: The student near the architect mentioned the dancers who supported the artist. Intervention (in the tokens for “near the architect”; simplified): – {subj_pp:near the architect} + {obj_rc_obj_pp:near the architect}	0.892 ± 0.009
---	---	---------------

Intended output change:
 The student near the architect mentioned the dancers who supported the artist. → The student mentions the dancers who support the artist near the architect.

r-rc_move_keep: Move a relative clause from after one noun to after another, without needing the relative clause's verb to change number	Original prompt: The popular photographer behind the happy doctor mentioned the famous banker who the farmer avoided. Intervention (in the tokens for “who the farmer avoided”; simplified): – {obj_rc:who the farmer avoided} + {subj_rc:who the farmer avoided}	0.910 ± 0.020
--	--	---------------

Intended output change:
 The popular photographer behind the happy doctor mentions the famous banker who the farmer avoids. → The popular photographer behind the happy doctor who the farmer avoids mentions the famous banker.

	r-rc_move_swap: Move a relative clause from after one noun to after another, requiring the relative clause's verb to change number	Original prompt: The helpful cyclists who visited the librarians mentioned the polite intern. Intervention (in the tokens for “who visited the librarians”; simplified): – {subj_rc:who visited the librarians} + {obj_rc:who visited the librarians} Intended output change: The helpful cyclists who visit the librarians mention the polite intern. → The helpful cyclists mention the polite intern who visits the librarians.	0.958 ± 0.017
Question formation	r-adj_move: Move an adjective from before one noun to before another	Original prompt: The inventor by the happy athlete could mention the musician. Intervention (in the “happy” token): – {subj_pp_adj:happy} + {subj_adj:happy} Intended output change: Could the inventor by the happy athlete mention the musician? → Could the happy inventor by the athlete mention the musician?	0.976 ± 0.018
	r-rc_arg_move: Change a subject relative clause to an object relative clause or vice versa	Original prompt: The spy who can encourage the important carpenter next to the philosopher would advise the writer. Intervention (in the tokens for “the important carpenter next to the philosopher”; simplified): – {subj_rc_obj:the important carpenter next to the philosopher} + {subj_rc_subj:the important carpenter next to the philosopher} Intended output change: Would the spy who can encourage the important carpenter next to the philosopher advise the writer? → Would the spy who the important carpenter next to the philosopher can encourage advise the writer?	0.962 ± 0.009

r-pp_move:	Original prompt:	0.912 ± 0.017
Move a prepositional phrase from after one noun to after another	The nurse can surprise the intern near the happy farmer.	

Intervention (in the tokens for “near the happy farmer”; simplified):

– {obj_pp:near the happy farmer} + {subj_pp:near the happy farmer}

Intended output change:

Can the nurse surprise the intern near the happy farmer? → Can the nurse near the happy farmer surprise the intern?

r-rc_move:	Original prompt:	0.886 ± 0.030
Move a relative clause from after one noun to after another	The famous baker could advise the sailor who can mention the careful director.	

Intervention (in the tokens for “who can mention the careful director”; simplified):

– {obj_rc:who can mention the careful director} + {subj_rc:who can mention the careful director}

Intended output change:

Could the famous baker advise the sailor who can mention the careful director? → Could the famous baker who can mention the careful director advise the sailor?

	Original	Edit to make
The	{subj_det-subj_det:The}	None
clever	{subj_det-subj_adj:The, subj_adj-subj_adj:clever}	- subj_det-subj_adj:The + subj_det-obj_adj:The - subj_adj-subj_adj:clever + obj_adj-obj_adj:clever + subj_n-obj_adj:spy + verb-obj_adj:helped + obj_det-obj_adj:the
spy	{subj_det-subj_n:The, subj_adj-subj_n:clever, subj_n-subj_n:spy}	- subj_adj-subj_n:clever
helped	{subj_det-verb:The, subj_adj-verb:clever, subj_n-verb:spy, verb-verb:helped}	- subj_adj-verb:clever
the	{subj_det-obj_det:The, subj_adj-obj_det:clever, subj_n-obj_det:spy, verb- obj_det:helped, obj_det- obj_det:the}	- subj_adj-obj_det:clever
poet	{subj_det-obj_n:The, subj_adj-obj_n:clever, subj_n-obj_n:spy, verb-obj_n:helped, obj_det-obj_n:the, obj_n- obj_n:poet}	- subj_adj-obj_n:clever + obj_adj-obj_n:clever
.	{subj_det-punct:The, subj_adj-punct:clever, subj_n-punct:spy, verb-punct:helped, obj_det- punct:the, obj_n-punct:poet, punct-punct:}.	- subj_adj-punct:clever + obj_adj-punct:clever

Table 3: Representational edits that would be made with the goal of moving *clever* from the subject to the object.

Task	Intervention type	Target acc.	Intervention (all)	Intervention (other)	Intervention (self)	Intervention (replace)
Arithmetic	f-number	0.990	0.978 ± 0.009	0.048 ± 0.009	0.858 ± 0.017	0.322 ± 0.054
	r-number_swap	1.000	0.978 ± 0.017	0.028 ± 0.026	0.875 ± 0.022	0.916 ± 0.028
	r-neg_move	1.000	0.982 ± 0.017	0.006 ± 0.011	0.976 ± 0.018	0.990 ± 0.000
Syllogisms	f-content	0.850	0.836 ± 0.018	0.026 ± 0.046	0.570 ± 0.174	0.000 ± 0.000
	f-function	0.770	0.592 ± 0.026	0.610 ± 0.058	0.038 ± 0.009	0.036 ± 0.023
	r-noun_swap	0.520	0.422 ± 0.026	0.424 ± 0.023	0.416 ± 0.011	0.488 ± 0.009
Code execution	f-list	0.990	0.990 ± 0.000	0.000 ± 0.000	0.974 ± 0.011	0.010 ± 0.000
	f-affix	1.000	1.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	0.040 ± 0.000
	f-rep	1.000	1.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	0.498 ± 0.071
	f-arg	0.990	0.960 ± 0.020	0.000 ± 0.000	0.952 ± 0.017	0.968 ± 0.026
	f-call	1.000	0.950 ± 0.014	0.000 ± 0.000	0.716 ± 0.041	0.958 ± 0.022
	r-list_move	1.000	0.950 ± 0.032	0.120 ± 0.037	0.290 ± 0.060	0.552 ± 0.046
Passivization	f-content	0.890	0.896 ± 0.011	0.000 ± 0.000	0.892 ± 0.009	0.052 ± 0.022
	r-adj_move	0.950	0.934 ± 0.018	0.584 ± 0.018	0.014 ± 0.018	0.026 ± 0.011
	r-rc_arg_move	0.880	0.856 ± 0.030	0.348 ± 0.048	0.224 ± 0.033	0.360 ± 0.014
	r-pp_move	0.920	0.696 ± 0.023	0.010 ± 0.000	0.236 ± 0.011	0.510 ± 0.020
	r-rc_move	0.940	0.638 ± 0.017	0.006 ± 0.011	0.356 ± 0.033	0.382 ± 0.017
Tense reinflection	f-content	0.980	0.950 ± 0.000	0.000 ± 0.000	0.940 ± 0.000	0.024 ± 0.033
	f-noun_1change	0.980	0.988 ± 0.009	0.000 ± 0.000	0.978 ± 0.009	0.010 ± 0.000
	f-noun_2change	0.990	0.990 ± 0.000	0.000 ± 0.000	0.970 ± 0.000	0.010 ± 0.000
	r-adj_move	0.980	0.980 ± 0.000	0.340 ± 0.032	0.034 ± 0.011	0.054 ± 0.018
	r-rc_arg_move_keep	0.960	0.986 ± 0.011	0.126 ± 0.061	0.526 ± 0.033	0.736 ± 0.033
	r-rc_arg_move_swap	0.990	0.964 ± 0.011	0.136 ± 0.033	0.526 ± 0.018	0.756 ± 0.023
	r-pp_move	0.990	0.892 ± 0.009	0.020 ± 0.000	0.190 ± 0.028	0.494 ± 0.041
	r-rc_move_keep	0.980	0.910 ± 0.020	0.086 ± 0.033	0.172 ± 0.017	0.368 ± 0.017
	r-rc_move_swap	1.000	0.958 ± 0.017	0.064 ± 0.054	0.176 ± 0.023	0.660 ± 0.020
Question formation	f-content	1.000	0.986 ± 0.011	0.000 ± 0.000	0.976 ± 0.011	0.078 ± 0.022
	r-adj_move	0.970	0.976 ± 0.018	0.466 ± 0.023	0.012 ± 0.017	0.020 ± 0.000
	r-rc_arg_move	0.950	0.962 ± 0.009	0.422 ± 0.073	0.184 ± 0.046	0.460 ± 0.028
	r-pp_move	0.970	0.912 ± 0.017	0.032 ± 0.017	0.266 ± 0.030	0.468 ± 0.038
	r-rc_move	0.950	0.886 ± 0.030	0.200 ± 0.058	0.198 ± 0.026	0.500 ± 0.014

Table 4: Detailed accuracies for causal interventions on GPT-OSS in symbolic domains. The intervention results show the mean across 5 reruns, plus or minus 2 standard deviations.

Task	Intervention type	Target acc.	Intervention (all)	Intervention (other)	Intervention (self)	Intervention (replace)
Arithmetic	f-number	0.990	0.980 $\pm$ 0.000	0.048 $\pm$ 0.009	0.878 $\pm$ 0.017	0.342 $\pm$ 0.054
	r-number_swap	1.000	0.978 $\pm$ 0.017	0.028 $\pm$ 0.026	0.875 $\pm$ 0.022	0.916 $\pm$ 0.028
	r-neg_move	1.000	0.982 $\pm$ 0.017	0.006 $\pm$ 0.011	0.976 $\pm$ 0.018	0.990 $\pm$ 0.000
Syllogisms	f-content	0.850	0.948 $\pm$ 0.022	0.040 $\pm$ 0.057	0.660 $\pm$ 0.202	0.010 $\pm$ 0.000
	f-function	0.770	0.718 $\pm$ 0.033	0.700 $\pm$ 0.032	0.048 $\pm$ 0.009	0.046 $\pm$ 0.023
	r-noun_swap	0.520	0.902 $\pm$ 0.026	0.906 $\pm$ 0.023	0.896 $\pm$ 0.011	0.964 $\pm$ 0.027
Code execution	f-list	0.990	1.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.984 $\pm$ 0.011	0.010 $\pm$ 0.000
	f-affix	1.000	1.000 $\pm$ 0.000	0.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.040 $\pm$ 0.000
	f-rep	1.000	1.000 $\pm$ 0.000	0.000 $\pm$ 0.000	1.000 $\pm$ 0.000	0.498 $\pm$ 0.071
	f-arg	0.990	0.952 $\pm$ 0.026	0.000 $\pm$ 0.000	0.944 $\pm$ 0.018	0.976 $\pm$ 0.027
	f-call	1.000	0.958 $\pm$ 0.017	0.006 $\pm$ 0.011	0.724 $\pm$ 0.041	0.966 $\pm$ 0.030
	r-list_move	1.000	0.950 $\pm$ 0.032	0.120 $\pm$ 0.037	0.290 $\pm$ 0.060	0.552 $\pm$ 0.046
Passivization	f-content	0.890	0.930 $\pm$ 0.020	0.000 $\pm$ 0.000	0.930 $\pm$ 0.014	0.052 $\pm$ 0.022
	r-adj_move	0.950	0.936 $\pm$ 0.027	0.588 $\pm$ 0.030	0.014 $\pm$ 0.018	0.026 $\pm$ 0.011
	r-rcargmove	0.880	0.884 $\pm$ 0.023	0.368 $\pm$ 0.036	0.248 $\pm$ 0.026	0.330 $\pm$ 0.014
	r-pp_move	0.920	0.712 $\pm$ 0.022	0.020 $\pm$ 0.000	0.246 $\pm$ 0.011	0.516 $\pm$ 0.027
	r-rc_move	0.940	0.658 $\pm$ 0.017	0.006 $\pm$ 0.011	0.368 $\pm$ 0.030	0.400 $\pm$ 0.014
Tense reinflection	f-content	0.980	0.968 $\pm$ 0.009	0.000 $\pm$ 0.000	0.960 $\pm$ 0.000	0.024 $\pm$ 0.033
	f-noun_1change	0.980	0.988 $\pm$ 0.009	0.000 $\pm$ 0.000	0.998 $\pm$ 0.009	0.010 $\pm$ 0.000
	f-noun_2change	0.990	1.000 $\pm$ 0.000	0.000 $\pm$ 0.000	0.980 $\pm$ 0.000	0.010 $\pm$ 0.000
	r-adj_move	0.980	0.990 $\pm$ 0.000	0.340 $\pm$ 0.032	0.034 $\pm$ 0.011	0.054 $\pm$ 0.018
	r-rc_arg_move_keep	0.960	0.966 $\pm$ 0.011	0.126 $\pm$ 0.061	0.526 $\pm$ 0.033	0.710 $\pm$ 0.024
	r-rc_arg_move_swap	0.990	0.954 $\pm$ 0.011	0.136 $\pm$ 0.033	0.526 $\pm$ 0.018	0.762 $\pm$ 0.026
	r-pp_move	0.990	0.882 $\pm$ 0.009	0.020 $\pm$ 0.000	0.190 $\pm$ 0.028	0.504 $\pm$ 0.041
	r-rc_move_keep	0.980	0.910 $\pm$ 0.020	0.086 $\pm$ 0.033	0.164 $\pm$ 0.018	0.368 $\pm$ 0.017
Question formation	r-rc_move_swap	1.000	0.958 $\pm$ 0.017	0.064 $\pm$ 0.054	0.176 $\pm$ 0.023	0.660 $\pm$ 0.020
	f-content	1.000	0.986 $\pm$ 0.011	0.000 $\pm$ 0.000	0.976 $\pm$ 0.011	0.078 $\pm$ 0.022
	r-adj_move	0.970	0.986 $\pm$ 0.018	0.474 $\pm$ 0.023	0.012 $\pm$ 0.017	0.020 $\pm$ 0.000
	r-rc_arg_move	0.950	0.988 $\pm$ 0.009	0.376 $\pm$ 0.067	0.184 $\pm$ 0.046	0.440 $\pm$ 0.028
	r-pp_move	0.970	0.900 $\pm$ 0.014	0.032 $\pm$ 0.017	0.266 $\pm$ 0.030	0.476 $\pm$ 0.039
	r-rc_move	0.950	0.904 $\pm$ 0.046	0.226 $\pm$ 0.039	0.188 $\pm$ 0.026	0.508 $\pm$ 0.017

Table 5: Detailed agreements for causal interventions on GPT-OSS in symbolic domains. The intervention results show the mean across 5 reruns, plus or minus 2 standard deviations.