

import pandas as np: the numpy strikes back

Haotian Yang

Bocconi University

haotian.yang@studbocconi.it

Abstract

Python code imports numpy as np so consistently that, to a language model, the alias and the library are almost the same thing. We ask what a model does when a file breaks the convention, as in `import pandas as np`. Using templated minimal pairs, we score how strongly a model prefers the conventional library’s methods at the use site and compare the swapped binding with a control that binds the same library to an alias with no convention (zz). Across nine open models from 0.5B to 8B parameters, the swap shifts the preference toward the conventional library by 6.8 nats relative to the control (95% CI 4.6 to 9.1). The shift is larger for more frequent conventions, stays flat out to 8,192 tokens, and reaches generated code, where 74% of swapped completions call an attribute the bound library lacks. DeepSeek-V4-Pro makes the same mistake out to 128k tokens even though it names the bound library correctly when asked, and its thinking mode removes the error. In two Qwen2.5 models, activation patching and the logit lens show that the pull travels with the alias token at the use site. Patching the import line barely moves it.

1 Introduction

A few import aliases are close to universal in Python, where numpy is imported as np, pandas as pd, and matplotlib.pyplot as plt. Public code follows this convention so consistently that a language model trained on it learns the alias and the library almost as one unit. The binding is still an ordinary line of code, and a program is free to change it. After `import pandas as np`, the name np refers to pandas, so `np.DataFrame` works and `np.arange` raises an `AttributeError`. A model that reads the file should expect pandas methods after np, and a model that goes by habit will expect numpy’s. This paper measures which of the two happens, how strongly, and where in the network the decision is made.

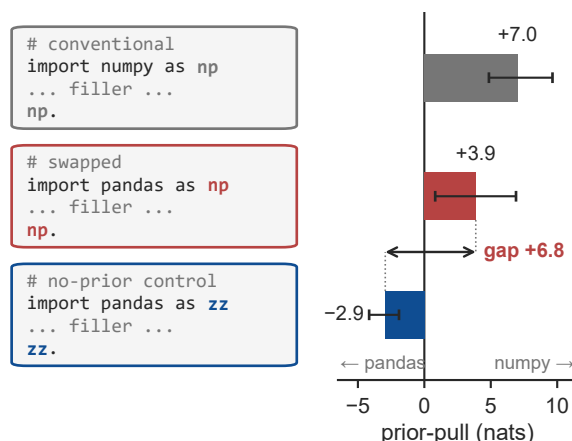


Figure 1: Three binding conditions for the alias np. Each prompt imports a library, adds filler code, and stops at the use site. The bars give the mean prior-pull there, the log-probability of the methods of the alias’s conventional library minus that of the other library’s (numpy minus pandas in this example), pooled over nine models, six alias pairs, and all distances, with 95% CIs. After `import pandas as np` the models still favor numpy. With a nonce alias they follow the binding. The difference between these two conditions, +6.8 nats, is the pull of the alias name.

The question is a small and checkable instance of a broader problem in NLP, the conflict between what a model reads in its input and what it learned in training (Xu et al., 2024). Question-answering models often keep a memorized answer after the supporting passage is edited to name a different entity (Longpre et al., 2021). Work on retrieval-augmented generation asks the same question about retrieved evidence, and Hagström et al. (2025) show that synthetic conflicts can overstate how much models use their context. Models also fall back on the usual meaning of an operator after a prompt redefines it (Thota et al., 2026). Code makes this kind of conflict unusually clean. The correct answer is fixed by the language, the context that states it is a single import line, and the

competing convention has a frequency that can be counted in public repositories. The setting also differs from work on renamed identifiers, where names are arbitrary and a model should ignore them (Ramakrishnan et al., 2022). A canonical alias is a name whose meaning the training data has almost settled.

We measure the pull with templated minimal pairs (Figure 1). Each prompt imports a library, continues with filler code that never mentions the alias, and ends at the point where the alias is used. There we compare how much probability the model gives to methods of the alias’s conventional library and to methods of the library it is actually bound to. The key comparison is between the swapped binding (`import pandas as np`) and a no-prior control (`import pandas as zz`), which binds the same library to an alias that has no convention behind it. Both conditions require the model to carry the same binding over the same distance, so the difference between them isolates the effect of the alias name. We then look inside two Qwen2.5 models with the logit lens (nostalgebraist, 2020), activation patching (Vig et al., 2020; Meng et al., 2022), and head-level attribution in the style of mechanistic studies of knowledge conflict (Yu et al., 2023; Jin et al., 2024).

In short, the convention wins far more often than it should, and the reason is not that the model fails to read the import. Our contributions are the following.¹

- A minimal-pair probe whose no-prior control separates the pull of a conventional alias from general difficulty with bindings, run on nine open models and one hosted frontier model.
- Evidence that the pull appears in every model, grows with the convention’s corpus frequency, does not change with distance, and produces calls that fail.
- A mechanistic account on two Qwen2.5 models. The bound library’s answer is overwritten late, the difference rides on the alias token at the use site, and the heads that fetch the binding are diverted.
- Mitigation probes showing that head ablation and a restating comment barely help, while ex-

plicit reasoning in the frontier model removes the error.

2 Setup

2.1 Stimuli and conditions

Each stimulus is a short Python prefix. It begins with an import line, continues with neutral filler code, and ends at the use site `alias.`, where the model has to produce a method name. For two libraries A and B , where a is the canonical alias of A , we build three conditions. The *conventional* condition imports A as a , which is the normal case. The *swapped* condition imports B as a , so the convention points to A while the binding says B . The *no-prior* condition imports B under a nonce alias such as `zz`. Swapped and no-prior share the bound library and the filler length and differ only in the alias name. The filler never mentions the alias, so the binding has to be carried from the import line to the use site. We generate the stimuli from templates instead of sampling real code. This keeps the conditions matched and avoids prompts that could occur verbatim in training data.

2.2 Measuring the pull

At the use site we score a few continuation strings for each library, for example `array(` and `arange(` for numpy and `DataFrame(` and `read_csv(` for pandas. Names that several libraries share, such as `sum` or `mean`, are left out (Appendix B). Method names often span several tokens, so each string is scored by the teacher-forced log-probability of all of its tokens. The *prior-pull* is the log-sum-exp over the conventional library’s strings minus the log-sum-exp over the other library’s strings. It is positive when the model favors the conventional library. Our main statistic is the difference-in-differences (DiD) gap, the prior-pull in the swapped condition minus the prior-pull in the no-prior condition.

2.3 Pairs, distances, and models

We use six swap pairs whose aliases span three levels of convention strength: very common (`np`, `pd`), common (`torch`, `plt`), and rare (`sk`, `xgb`). The filler places the import 0, 512, or 2,048 tokens before the use site, counted with each model’s own tokenizer, and 8,192 tokens for the models run on CPU. We test nine configurations of eight open checkpoints between 0.5B and 8B parameters (Table 1). They are Qwen2.5 0.5B and 1.5B

¹Code, stimuli, scored outputs, and figures are available at <https://github.com/soray42/pandas-as-pd> under the MIT license. The model-generated naturalistic contexts are released as validated records with hashed prompts, together with the scripts that generated and validated them.

in base and instruct versions, Qwen2.5-Coder-7B, Llama-3.2 1B and 3B, and Llama-3.1-8B (Qwen et al., 2024; Grattafiori et al., 2024). The four small Qwen2.5 checkpoints run in fp16 on a GPU. The other models run as quantized GGUF files on a CPU, which makes the 8,192-token bins feasible, and Qwen2.5-0.5B-Instruct is run both ways. The full run scores 1,683 prefixes. In a generation arm the models also decode 48 tokens greedily after alias. for 468 prompts, and we check whether the first attribute they access exists on the real bound library. If it does not, the call is broken and would raise an `AttributeError`. All intervals are 95% bootstrap intervals clustered on swap pair.

2.4 Mechanistic setup

For the analysis in Section 4 we load Qwen2.5-0.5B and 1.5B (base) in TransformerLens (Nanda and Bloom, 2022) and use the numpy/pandas pair. We build 30 base contexts at each of two depths, 0 and 512 tokens, and write each one in all three conditions. Within a base the three prompts differ only in the library name and the two alias tokens, all of which are single tokens, so every position lines up across conditions (Appendix C). Because the behavioral effect does not depend on distance, short contexts are enough. Layer-by-layer readouts need a metric that looks at a single position, so we use a proxy, the log-sum-exp of the final-position logits over the first token of each library’s continuations. On the same 180 prompts per model, the proxy ranks items much like the full prior-pull (Spearman 0.87 for both models, sign agreement 0.98 and 0.93).

3 Results

3.1 Open models

Swapping the alias moves every model toward the conventional library (Table 1). The overall DiD gap is +6.81 nats (95% CI [+4.56, +9.11]). It is positive for all nine models, and no model’s interval includes zero. The smallest gap is +5.17 for Llama-3.2-3B and the largest +7.56 for Qwen2.5-1.5B-Instruct. Put in terms of odds, the swap multiplies the ratio of conventional to bound probability mass by about 900 relative to the control.

The raw values show what the control contributes (Figure 1 and Appendix Table 4). Under the no-prior binding the models prefer the bound library by 2.94 nats, so they can track the binding when nothing competes with it. With the canonical

Model	Tune	DiD gap [95% CI]
Qwen2.5-0.5B	base	+6.70 [+4.15, +9.05]
Qwen2.5-0.5B	inst	+6.93 [+4.33, +9.19]
Qwen2.5-0.5B [†]	inst	+5.51 [+2.41, +9.66]
Llama-3.2-1B [†]	inst	+7.08 [+2.99, +10.0]
Qwen2.5-1.5B	base	+7.46 [+4.64, +10.0]
Qwen2.5-1.5B	inst	+7.56 [+4.78, +10.3]
Llama-3.2-3B [†]	inst	+5.17 [+2.65, +8.72]
Qwen2.5-Coder-7B [†]	inst	+6.33 [+3.31, +9.63]
Llama-3.1-8B [†]	inst	+7.33 [+3.73, +10.3]
Overall		+6.81 [+4.56, +9.11]

Table 1: DiD gap (swapped minus no-prior prior-pull, in nats) for each model, with 95% pair-clustered bootstrap intervals. Every interval excludes zero. [†]Quantized GGUF on CPU, 4-bit except for Llama-3.2-1B (8-bit). The other rows run in fp16 on a GPU.

alias this becomes a 3.87-nat preference for the conventional library, whose methods then carry about 48 times the probability mass of the bound library’s. The control is a measured baseline, not a neutral zero, and subtracting it also cancels differences between the two libraries’ method lists.

If the pull is learned from training data, more frequent conventions should pull harder. We counted each canonical import line, for example `import numpy as np`, in 150,000 Python files from the codeparrot corpus and used the log count as a continuous dose.² We then regressed the prior-pull on the swapped indicator, the dose, their interaction, and depth, with crossed random intercepts for pair and model, over all 1,122 swapped and no-prior items. The gap grows by 0.94 nats per natural-log unit of frequency (Figure 2). A bootstrap over pairs gives a 95% interval of [+0.69, +1.35], and 99% of resampled slopes are positive. These items come from only six alias pairs, so we read the slope as a clear direction with an imprecise size. The hosted model in Section 3.2 shows the same direction over twelve conventions.

Distance does not matter over the range we can test. Within each model the gap is roughly constant from 0 to 8,192 tokens, and no model’s gap in its longest bin exceeds its gap in its shortest bin with non-overlapping intervals (Appendix Figure 5). A single run of Qwen2.5-0.5B at 32,768 tokens gives a gap of +8.9 nats (swapped +9.0, no-prior +0.1), in line with its shorter bins. The pull is there as soon as the alias is used and does not build up

²<https://huggingface.co/datasets/codeparrot/codeparrot-clean>, counted by `scripts/corpus_freq.py`. When an alias equals its module name (`torch`), a plain `import` also establishes the binding and is counted.

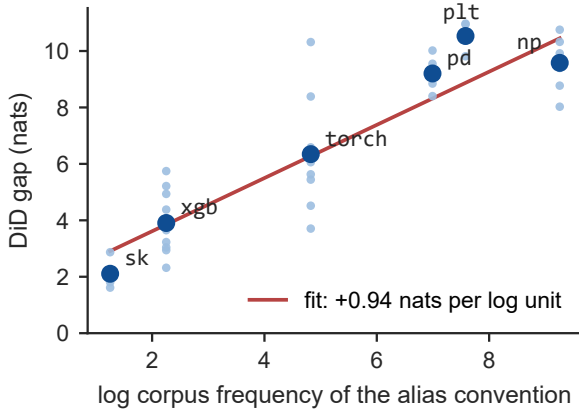


Figure 2: Dose response. The DiD gap for each alias pair (large dots) and for each pair and model (small dots), against the log frequency of the canonical import `LIB as alias` line in public Python. The line is the mixed-model fit, $+0.94$ nats per log unit (pair bootstrap 95% CI $[+0.69, +1.35]$).

with distance, unlike long-context failures in which retrieval degrades with position (Liu et al., 2024; Modarressi et al., 2025). Scale and instruction tuning do not remove it either. The per-model gaps show no trend between 0.5B and 8B (Appendix Figure 7), and the instruct models differ from their base versions by -0.08 nats on average over the two matched Qwen2.5 pairs.

The preference shows up in what the models write. In greedy generation after `alias.`, calls to the conventional library appear only in the swapped condition, in 18% of completions against 0% in the other two. The broken-call rate is 0.74 under the swap, 0.46 under the no-prior control, and 0.12 in the conventional condition (Appendix Figure 6). The share of conventional-library calls is modest compared with the size of the log-probability preference, but the calls that do appear fail under the real binding. For a coding assistant this means that a valid but unusual import can be overruled at the point of use, even with the import in view.

3.2 A hosted frontier model

DeepSeek-V4-Pro (DeepSeek-AI, 2026) has a context window of one million tokens, but its API does not allow scoring arbitrary continuations. We therefore test it with three behavioral tasks at the use site and keep the comparison between swapped and no-prior (Table 2). In a forced choice the model picks one of two methods, one from the conventional library, which does not exist under the binding, and one from the bound library. In free generation it completes `alias.`, and in a verbal task it

Probe	direct	think
Forced choice, prior-method rate		
conventional (ceiling)	0.92	1.00
swapped	0.65	0.00
no-prior (control)	0.36	0.00
swapped – no-prior	+0.30	0.00
Verbal recognition, swapped	1.00	1.00

Table 2: DeepSeek-V4-Pro behavioral probe. In the forced choice the direct model picks the conventional library’s non-existent method far more often under the swap than under the no-prior control (difference $+0.30$, 95% CI $[+0.06, +0.61]$), yet it names the bound library correctly every time. Thinking mode removes the forced-choice pull. The swapped forced-choice rate is flat from 0 to 128k tokens (0.50, 0.75, 0.58, 0.75, and 0.50 at 0, 2k, 8k, 32k, and 128k, with $n=4$ at 128k). Direct-mode n is 24 to 46 per condition.

names the library that the alias refers to. In direct mode, without reasoning, the model picks the non-existent conventional method on 65% of swapped items and 36% of no-prior items, a difference of $+0.30$ (95% CI $[+0.06, +0.61]$). Asked in words, it names the bound library correctly on every item, swapped items included. The model can report the binding but does not use it when it chooses a method. Free generation agrees, with broken-call rates of 0.69 (swapped), 0.46 (no-prior), and 0.05 (conventional), although the chat format makes this arm noisier. On the two most common conventions the swapped forced-choice rate stays between 0.50 and 0.75 from 0 to 131,072 tokens of context, far beyond what the open models can process.

Explicit reasoning removes the error. In thinking mode (DeepSeek-AI et al., 2025; Wei et al., 2022) the swapped forced-choice rate drops to 0.00, while the conventional rate is 1.00 and verbal recognition stays at 100%. A note stating the binding, placed before the question, helps much less. It lowers the prior-choice rate on np from 1.00 to 0.75 and on pd from 0.88 to 0.50. We also sampled the forced choice at temperature 0.7, eight draws per item, for twelve conventions. The prior-choice rate rises with the convention’s corpus frequency (Pearson $r = 0.59$, positive slope in 98% of bootstrap resamples) and is highest for np and pd, which exceed the control by 0.79 and 0.83.

3.3 Realistic contexts and other nonce aliases

Templated prompts are a controlled but artificial instrument, so we also score model-generated snippets whose surrounding task clearly needs the bound library, for all twelve conventions. The bind-

ing conditions are applied to these snippets by a program, and every prompt is hashed. Both local models we run on them, Qwen2.5-Coder-7B and Qwen2.5-0.5B, show the pull, with DiD gaps of +5.7 and +7.6 nats and intervals that exclude zero. The largest gaps again come from the most frequent conventions, and the completions contain idioms that fail under the binding, such as a pandas Series where numpy is bound. DeepSeek-V4-Pro in direct mode breaks 9.2% of swapped completions against 3.1% under the control (+6.2 points, 95% CI [+0.0, +12.3]), and thinking mode brings the swapped rate down to 0.8%. Its absolute error rate is far below that of the small models.

The templated runs use a single nonce alias, `zz`, so the control could in principle reflect something specific to that token. The naturalistic arm and the sampled forced choice both use three nonce aliases (`zz`, `qx`, and `vv`), which allows a check. In the naturalistic arm the gap hardly changes with the nonce. It is +5.6, +5.8, and +5.7 nats for Qwen2.5-Coder-7B and +7.7, +7.1, and +8.0 for Qwen2.5-0.5B, with every interval above zero. In the forced choice the no-prior prior-choice rate, averaged over the twelve conventions, is 0.20, 0.19, and 0.21 for the three nonces. The control does not depend on the choice of `zz`.

4 Where the prior acts

The behavioral results show that the pull exists, grows with frequency, and does not decay. They do not show how the convention wins. Mechanistic studies of knowledge conflict describe heads that recall a memorized answer and heads that copy from the context, and they find that the competition between the two is settled late in the network (Yu et al., 2023; Jin et al., 2024; Ortu et al., 2024). Two accounts fit our setting. Either the bound library’s answer never forms, or it forms in the middle layers and is overwritten near the output. We test them on Qwen2.5-0.5B (24 blocks) and 1.5B (28 blocks), using the numpy/pandas pair and the proxy metric from Section 2.

4.1 Layerwise readout and patching

We decode the residual stream at the final position after every block through the final norm and the unembedding (nostalgebraist, 2020; Geva et al., 2022) and follow the proxy-pull through the network (Figure 3). The swapped trajectory follows the conventional one at every layer, in both models

and at both depths. After block 2 the two are on average 0.25 to 0.55 nats apart, while swapped and no-prior are 3.8 to 4.6 nats apart. The trajectory depends on which alias is used and hardly at all on what it is bound to. The bound library does take the lead somewhere after block 4 in 90 to 100% of swapped items, but only weakly. The lowest point of a swapped trajectory averages -0.58 nats in 0.5B, against -8.31 under the no-prior binding, and -6.70 against -13.86 in 1.5B, an attenuation of 93% and 52%. The decision comes late. In 1.5B the conditions split in blocks 22 to 24, where the no-prior trajectory falls to -14.9 as the model retrieves the binding, while the conventional and swapped trajectories rise to between $+4$ and $+6$. The last block at which the bound library leads is on average block 23 of 28 in 1.5B and 19.3 to 19.6 of 24 in 0.5B, and every swapped item ends with a positive proxy-pull. In 1.5B the bound answer forms and is then overwritten. In 0.5B it barely forms.

The logit lens only shows correlations. To test causal roles we patch activations between runs (Vig et al., 2020; Meng et al., 2022; Zhang and Nanda, 2024). Within a base, the swapped and no-prior prompts contain the same tokens except for the alias, so copying activations from one into the other isolates the effect of the alias name. We copy the residual stream at one layer and one group of positions from a no-prior run into the matching swapped run and measure the fraction of the proxy-pull gap between the two runs that the patch restores. The position groups are the import line, the alias at the use site, the final position, and the filler, which serves as a negative control. In total we run 24,960 patches.

Figure 4 shows the result at depth 0. Patching the use-site alias restores 0.79 of the gap in 0.5B and 0.61 in 1.5B at layer 0, where the patch amounts to swapping in the other token. Its effect then fades and reaches zero at layer 20 in 0.5B and layer 22 in 1.5B. The final position takes over as the alias position fades. In 0.5B it rises gradually and restores the full gap by layer 19. In 1.5B it jumps from 0.24 at layer 18 to 0.97 at layer 22, in the same blocks where the logit-lens trajectories split. The import line never restores more than 0.08 of the gap, and the filler at depth 512 restores none ($|f| \leq 0.02$). The convention therefore does not act by corrupting what the model reads from the import. The alias token carries it from the embedding upward and hands it to the final position, where it is written

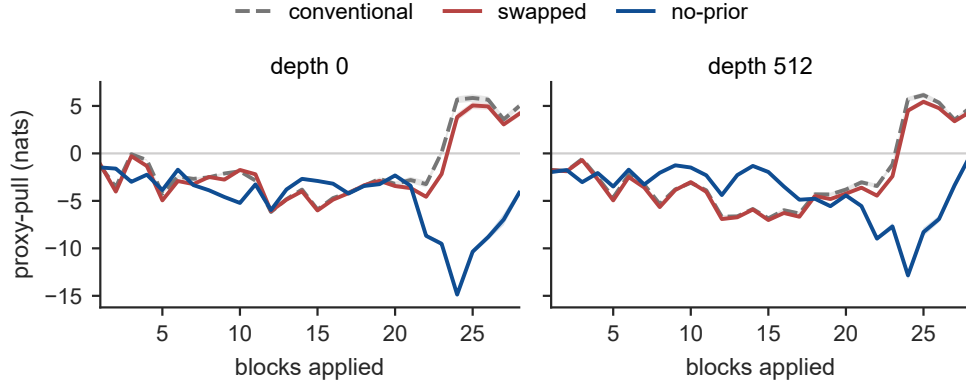


Figure 3: Logit-lens proxy-pull after each block of Qwen2.5-1.5B, mean over items with 95% bootstrap bands (the embedding-only checkpoint is omitted). The swapped trajectory follows the conventional one, not the no-prior one, at every block. The conditions split in blocks 22 to 24, where no-prior retrieves the binding (lowest point -14.9) and swapped resolves to the conventional library. Qwen2.5-0.5B shows the same shape with a shallower swapped trough (Appendix Figure 9).

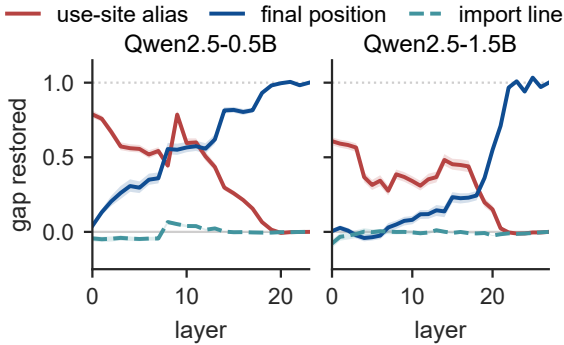


Figure 4: Activation patching from a no-prior run into the matching swapped run at depth 0. Each line is the fraction of the proxy-pull gap restored by patching one group of positions at one layer, averaged over 30 bases with 95% bootstrap bands. The use-site alias carries the difference from layer 0 and hands it to the final position in the late layers. The import line restores almost none of it. The filler is empty at depth 0 and restores none at depth 512. Appendix D describes the reverse direction.

into the output in the last blocks.

4.2 Attention heads

Next we look for the heads that should carry the binding. At the final position we measure each head’s attention to the import line and its direct contribution to the output along the difference between the two libraries’ mean unembedding vectors (Elhage et al., 2021). Raw attention to the import is hard to interpret, because the import sits at the start of the sequence, where many heads put attention by default (Xiao et al., 2024). Averaged over all heads that attend to the import, attention does not fall under the swap. We therefore compare each head

Head	import attn		DLA	
	no-pr	swap	no-pr	swap
0.5B L17H1	0.61	0.09	-0.24	-0.17
1.5B L21H5	0.86	0.29	-0.15	+0.02
1.5B L22H8	0.52	0.11	-0.07	+0.21

Table 3: Signature heads at depth 512. Attention from the final position to the import line, and direct logit attribution (DLA) along the proxy direction, where negative values promote the binding. Under the canonical alias the same heads stop attending to the import and stop promoting the binding, or reverse. Depth-0 values agree (Appendix Table 6). L21H5 is selected by its attention contrast, not by its swapped-side attribution, which the diversion itself drives to zero.

with itself across conditions, which keeps token positions fixed.

The heads with the largest contrast that also affect the output share one pattern (Table 3). Under the nonce alias they attend to the import and push toward the bound library. Under np their attention to the import collapses and their push toward the binding weakens or reverses. The clearest case is head L22H8 of 1.5B. It gives 0.52 to 0.70 of its attention to the import under zz but only 0.11 to 0.23 under np, and under np it becomes the head that pushes hardest toward numpy, inside the late window where the output is decided. In the terms of Yu et al. (2023) and Jin et al. (2024), the context heads exist and work. The alias token changes what they look for, so they stop reading the import, and the late layers settle on the alias’s usual library.

4.3 Mitigations

If the late write lived in a few heads, removing them should weaken the pull. We zero-ablated the one, three, or five heads with the largest positive attribution, and the pull did not drop. It stayed flat in 0.5B (swapped $+2.60 \rightarrow +2.52$ at best) and rose slightly in 1.5B ($+4.27 \rightarrow +4.74$ at $k=1$), and the conventional condition moved at least as much (Appendix Table 7). Direct attribution measures what a head contributes, not what happens when the head is gone, and the mismatch fits the self-repair reported in other ablation studies (McGrath et al., 2023; Rushing and Nanda, 2024). The late write is spread over many components. The only intervention that changes the behavior is replacing the alias representation at the use site, which restores 0.6 to 1.2 of the gap when patched (Appendix D).

Text-level fixes behave as this account predicts. We measured them with the full behavioral metric on four Qwen2.5 checkpoints (Appendix Table 8). A comment restating the binding just before the use site (# note: np is pandas in this file) reduces the pull by 1.5 to 2.6 nats, with every interval excluding zero, but it still leaves 6.0 to 7.3 nats of a gap of about 10. An instruction above the import to resolve aliases by the imports moves the instruct models by at most 0.85 nats. Neither intervention touches the alias token that carries the pull. The results also match the hosted model, where a note helped a little and only explicit reasoning removed the error. On the rare pair, sk bound to xgboost, the swapped condition already follows the binding (-1.2 to -3.3 nats), which fits the frequency trend.

5 Related Work

Our setting is a code instance of knowledge conflict, in which the input contradicts what a model learned in training (Xu et al., 2024). In question answering, models often return a memorized answer after the supporting passage is edited to name a different entity (Longpre et al., 2021). Studies of retrieval-augmented generation measure how much models rely on retrieved context, and Hagström et al. (2025) find that synthetic conflict datasets exaggerate context properties that are rare in real retrieved evidence and inflate estimates of context use. Thota et al. (2026) report that models revert to the standard meaning of an operator after a prompt redefines it, in Boolean algebra and circuit reasoning. We observe the same failure with library bindings in code, measure it as a graded preference

against a matched control, and relate it to how often the competing convention appears in public code.

Mechanistic studies of these conflicts find memory heads and context heads whose competition decides between a memorized and an in-context answer. Yu et al. (2023) control the outcome by scaling single heads, Jin et al. (2024) place the conflict at a late stage where the two streams are combined and reduce it by pruning heads, and Ortu et al. (2024) trace the competition between factual and counterfactual answers to a few heads and the attribute position. Our setting moves the conflict from facts stated in natural language to name bindings in code, where the context implies the answer through composition instead of stating it. We add two observations. The diversion is visible on the query side, since the same context heads stop attending to the import when the alias has a convention. And the single-head ablations that control factual recall in this earlier work did not reduce the pull in ours, which is consistent with self-repair (McGrath et al., 2023; Rushing and Nanda, 2024).

Work on the semantic robustness of code models asks them to stay invariant when identifiers are renamed (Ramakrishnan et al., 2022). That goal makes sense when names are arbitrary. A canonical alias is not arbitrary, and a model that ignored it entirely would throw away useful information. Our question is narrower, namely whether the model still defers to an explicit binding when the name and the binding disagree. Long-context benchmarks, finally, measure how well a model retrieves information placed at different positions (Liu et al., 2024; Modarressi et al., 2025). Our distance axis holds the conflict fixed and varies how far the import is from the use site. The pull does not decay over the range we test, which places it closer to knowledge conflict than to retrieval failure. For methods we follow templated-stimulus circuit analysis (Wang et al., 2023), logit-lens readouts (nostalgebraist, 2020; Belrose et al., 2023; Geva et al., 2022), and published practice for activation patching (Zhang and Nanda, 2024; Heimersheim and Nanda, 2024).

6 Conclusion

A canonical import alias is a token whose meaning the training data has almost fixed. When code rebinds such an alias, open models from 0.5B to 8B parameters, and a frontier model out to 128k tokens, keep favoring the conventional library. The

effect grows with how common the convention is, appears as soon as the alias is used, and is large enough to produce code that cannot run. Inside two small models the failure is not one of reading the import. The binding is available, and the model uses it when the alias has no convention. The alias token itself carries the convention to the output, and the heads that would fetch the binding look elsewhere. Restating the binding helps a little, and explicit reasoning fixes the error. Code is a setting in which the context should always win, because the interpreter always follows the import. That makes it a useful test bed for work on conflicts between context and memory.

Limitations

Our setting is narrow. We study one kind of binding, a canonical import alias rebound to another real library, and most experiments use templated code in which the swap is the only unusual thing. Real programs rarely rebound `np`, and other bindings, such as variables or arguments whose names suggest a type, may behave differently. As Hagström et al. (2025) show for retrieval, synthetic conflicts can misstate how models behave on real inputs. Our naturalistic arm uses model-generated contexts and serves as a check, not a full sweep. We also did not test whether earlier correct uses of the rebound alias in the same file, a stronger form of local evidence than a comment, would overcome the convention. That is a natural next experiment.

The frequency analysis rests on six alias pairs in the continuation-scoring runs and twelve in the forced choice, so the size of the slope is imprecise, and the corpus counts come from a 150,000-file sample of public Python, not from any model’s training data. The templated runs use one nonce alias, although three nonces gave indistinguishable controls in the other two arms. Distance reaches 8,192 tokens for all open models and 32,768 for one model, limited by context windows and hardware. The open models are 8B or smaller, and the one frontier model is probed behaviorally because its API does not support teacher forcing. The metric is a preference at a single use site, and conventional-library generations are modest in number, so the cost in multi-step coding is not measured here.

The mechanistic analysis covers two sizes of one model family on one pair at two depths, and it relies on a proxy metric validated at Spearman 0.87, so the layerwise conclusions are claims about that

proxy. The logit lens is correlational, patching is causal only at the level of residual-stream positions, and we do not identify a circuit. In the 1.5B model the hooked implementation’s absolute logits drift from the reference implementation at 512 tokens, so there we rely only on contrasts between conditions. Direct logit attribution is a linear, descriptive measure, and our ablations show that it does not predict what happens under intervention.

Ethics Statement

The study involves no human subjects and no personal data. All stimuli are synthetic or model-generated code, and the model-generated ones are marked as such. Model weights are used under their original licenses, and our code and data are released under the MIT license. Most errors we observe would raise an exception at run time, but a conventional name can also exist in the bound library (`pandas.array`, for example) and then run with different behavior. We report which mitigations help and which do not.

References

- Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. 2023. [Eliciting latent predictions from transformers with the tuned lens](#). *Preprint*, arXiv:2303.08112.
- DeepSeek-AI. 2026. [DeepSeek-V4: Towards highly efficient million-token context intelligence](#). Technical report.
- DeepSeek-AI, Daya Guo, and 1 others. 2025. [DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning](#). *arXiv preprint arXiv:2501.12948*.
- Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. [A mathematical framework for transformer circuits](#). *Transformer Circuits Thread*.
- Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. [Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45. Association for Computational Linguistics.
- Aaron Grattafiori and 1 others. 2024. [The Llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- Lovisa Hagström, Sara Vera Marjanovic, Haeun Yu, Arnav Arora, Christina Lioma, Maria Maistro, Pepa Atanasova, and Isabelle Augenstein. 2025. [A reality check on context utilisation for retrieval-augmented generation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19691–19730, Vienna, Austria. Association for Computational Linguistics.
- Stefan Heimersheim and Neel Nanda. 2024. [How to use and interpret activation patching](#). *arXiv preprint arXiv:2404.15255*.
- Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. 2024. [Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1193–1215, Bangkok, Thailand. Association for Computational Linguistics.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. [Entity-based knowledge conflicts in question answering](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Thomas McGrath, Matthew Rahtz, János Kramár, Vladimir Mikulik, and Shane Legg. 2023. [The hydra effect: Emergent self-repair in language model computations](#). *arXiv preprint arXiv:2307.15771*.
- Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. [Locating and editing factual associations in GPT](#). In *Advances in Neural Information Processing Systems*, volume 35.
- Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. 2025. [NoLiMa: Long-context evaluation beyond literal matching](#). In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*.
- Neel Nanda and Joseph Bloom. 2022. Transformerlens. <https://github.com/TransformerLensOrg/TransformerLens>.
- nostalgebraist. 2020. Interpreting GPT: the logit lens. <https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens>. LessWrong blog post, Aug. 31, 2020.
- Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. [In-context learning and induction heads](#). *Transformer Circuits Thread*.
- Francesco Ortú, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. 2024. [Competition of mechanisms: Tracing how language models handle facts and counterfactuals](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8420–8436, Bangkok, Thailand. Association for Computational Linguistics.
- Qwen, An Yang, Baosong Yang, Beichen Zhang, and 1 others. 2024. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Goutham Ramakrishnan, Jordan Henkel, Zi Wang, Aws Albarghouthi, Somesh Jha, and Thomas Reps. 2022. [Semantic robustness of models of source code](#). In *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 526–537.

- Cody Rushing and Neel Nanda. 2024. [Explorations of self-repair in language models](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 42836–42855. PMLR.
- Yogeswar Reddy Thota, Setareh Rafatirad, Houman Homayoun, and Tooraj Nikoubin. 2026. [When models ignore definitions: Measuring semantic override hallucinations in LLM reasoning](#). *arXiv preprint arXiv:2602.17520*.
- Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. [Investigating gender bias in language models using causal mediation analysis](#). In *Advances in Neural Information Processing Systems*, volume 33.
- Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. [Interpretability in the wild: a circuit for indirect object identification in GPT-2 small](#). In *The Eleventh International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. [Efficient streaming language models with attention sinks](#). In *The Twelfth International Conference on Learning Representations*.
- Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. [Knowledge conflicts for LLMs: A survey](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8541–8565.
- Qinan Yu, Jack Merullo, and Ellie Pavlick. 2023. [Characterizing mechanisms for factual recall in language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9924–9959, Singapore. Association for Computational Linguistics.
- Fred Zhang and Neel Nanda. 2024. [Towards best practices of activation patching in language models: Metrics and methods](#). In *The Twelfth International Conference on Learning Representations*.

Condition	Raw prior-pull [95% CI]
Conventional	+7.02 [+4.86, +9.65]
Swapped	+3.87 [+0.82, +6.90]
No-prior	-2.94 [-4.15, -1.91]
DiD gap	+6.81 [+4.56, +9.11]

Table 4: Raw prior-pull by condition (nats), pooled over pairs, models, and depths, with 95% pair-clustered bootstrap intervals ($n=561$ items per condition). The prior-pull is the log-probability preference for the pair’s conventional library over the other library at the use site. In the conventional condition the models prefer the correct library. Under no-prior they prefer the bound library, so they track the binding when no convention competes, and the swap turns this into a preference for the conventional library. The DiD gap subtracts the no-prior baseline, which cancels differences between the lexicons.

A Additional Behavioral Results

Table 4 gives the raw prior-pull by condition behind the DiD gap. Figure 5 shows the gap against distance for each model, Figure 6 the generation arm, and Figure 7 the gap against model size.

B Candidate Continuations and Versions

Table 5 lists the continuation sets, their mean token length under a reference tokenizer, and the package version at which we checked that every continuation is a real attribute or, for sklearn sub-modules, an importable subpackage. The six libraries above the rule are used in the continuation-scoring runs, and the six below extend the forced choice to a wider frequency range. Mean lengths are similar across libraries (1.0 to 2.8 tokens at the use site), so no library’s candidates are systematically longer, and summing teacher-forced log-probabilities scores multi-token names consistently. One pandas entry, groupby, is a DataFrame method and not a module-level function. We keep it as a typical pandas idiom but never use it as a forced-choice option, since those options must resolve at module level. Per-continuation records are in results/candidate_table.json.

C Mechanistic Setup and Validation

Alignment. For the numpy/pandas pair the three conditions of a base tokenize to the same length with the same spans. numpy and pandas are single tokens, and so are the aliases np and zz, both after as and at the use site. Each base fixes the variable on the use line, drawn from a pool of 30 names (x,

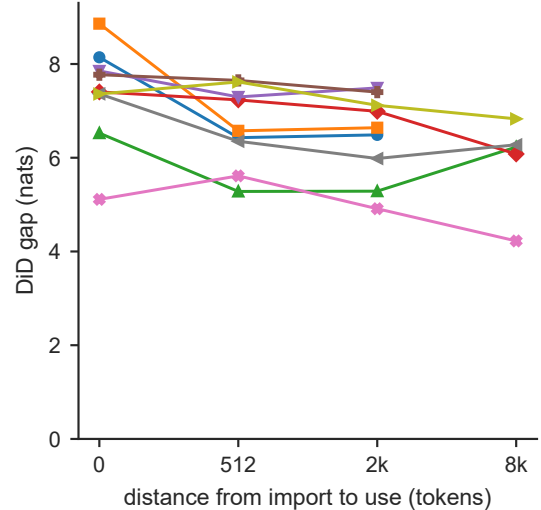


Figure 5: DiD gap against the distance from the import to the use site, one line per model. The gap stays positive and roughly flat from 0 to 8,192 tokens, and no model shows an increase from its shortest to its longest bin with non-overlapping intervals.

res, tmp, ...), and at depth 512 a filler seed. The conditions then differ only at the library token and the two alias positions, which we check element by element for every triple. This gives 30 bases per depth and 180 stimuli per model, the same for both models since they share a tokenizer.

Proxy lexicon. The proxy uses the first token of each of the six continuations per library, tokenized without a leading space because they follow the dot. No first token appears in both libraries’ sets, so nothing is dropped. The proxy-pull is the log-sum-exp of the final-position logits over the numpy set minus that over the pandas set, in this orientation for all three conditions. Adding a constant to all logits does not change it, so it can be computed on raw logits.

Validation. We load the models without weight processing (no norm folding or centering). The logit lens at the final checkpoint then reproduces the true final logits exactly (identical argmax, correlation 1.0), which anchors the basis of the lens. Against the full continuation-scored prior-pull on the same 180 prompts, computed with the scoring code of the behavioral runs, the proxy reaches Pear-

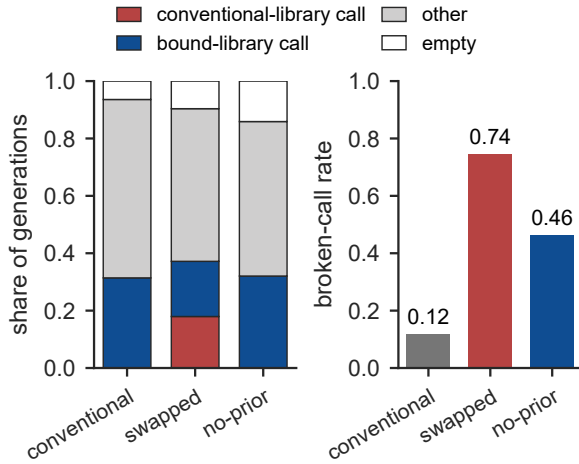


Figure 6: Generation arm. Left, the class of the first attribute accessed after alias. in each condition. Only the swapped condition produces conventional-library calls. Right, the broken-call rate, the share of generations whose first attribute does not exist on the bound library.

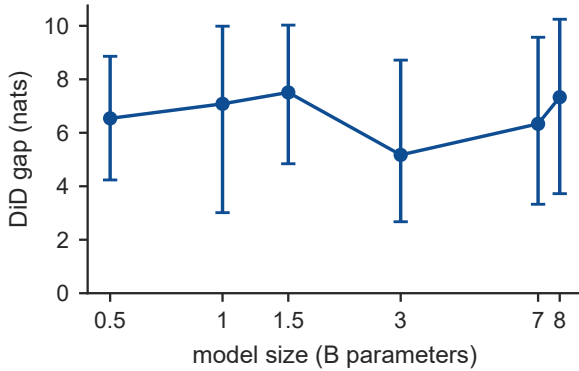


Figure 7: DiD gap against model size, pooled over the models of each size, with 95% pair-clustered bootstrap intervals. The gap shows no trend from 0.5B to 8B.

son 0.98 and 0.94 and Spearman 0.87 and 0.87 for 0.5B and 1.5B. On items where the full metric exceeds 1 nat in size ($n=160$ and 180), the two agree in sign on 98% and 93% of items (Figure 8). We fixed the thresholds for proceeding before running the layerwise experiments (Spearman at least 0.7 and sign agreement at least 0.9 for both models), and the validation gives identical statistics when run twice. The hooked and reference implementations agree for 0.5B (same argmax on 6 of 6 probes, mean largest log-probability difference 0.05 nats). For 1.5B they disagree on 2 of 6 probes, only at depth 512 and only between top candidates within 1 nat of each other, which reflects fp16 accumulation differences between eager attention and fused kernels. All layerwise quantities are contrasts be-

Library	Alias	n	tok	Version
numpy	np	6	1.0	2.2.6
pandas	pd	6	1.2	3.0.1
torch	torch	6	1.3	2.12.0.dev20260405
sklearn	sk	6	1.5	1.8.0
xgboost	xgb	6	2.2	3.2.0
matplotlib.pyplot	plt	6	1.0	3.10.8
seaborn	sns	6	2.5	0.13.2
networkx	nx	6	2.8	3.6.1
statsmodels.api	sm	5	2.0	0.14.6
sqlalchemy	sa	5	1.4	2.0.45
sympy	sym	6	1.5	1.14.0
plotly.express	px	6	2.8	6.3.1

Table 5: Continuation sets per library: number of strings, mean use-site token length (tiktoken c1100k), and the pinned package version used to verify them. The six libraries above the rule are used in the continuation-scoring runs. The six below extend the forced-choice dose response.

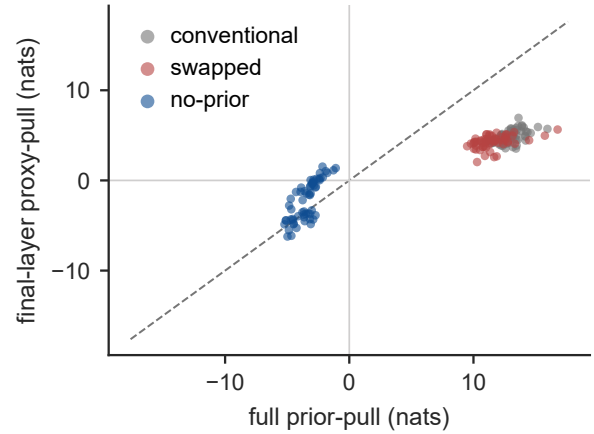


Figure 8: Proxy validation for Qwen2.5-1.5B. Final-layer proxy-pull against the full continuation-scored prior-pull on the same prompts, by condition (Spearman 0.87). The 0.5B scatter is tighter (Pearson 0.98).

tween conditions within one implementation, so this drift largely cancels.

D Additional Mechanistic Results

Trajectories. Figure 9 shows the logit-lens trajectories for 0.5B. The swapped trough is shallower than in 1.5B (mean per-item minimum -0.58 against -6.70), which is why we describe the 0.5B answer as barely formed. In 3 of 30 swapped items at depth 512 the bound library never leads after block 4.

Patching. In the reverse direction, from a swapped run into the matching no-prior run, the use-site alias restores 1.0 of the gap in 0.5B and 1.2 in 1.5B at layer 0. Transplanting the np representa-

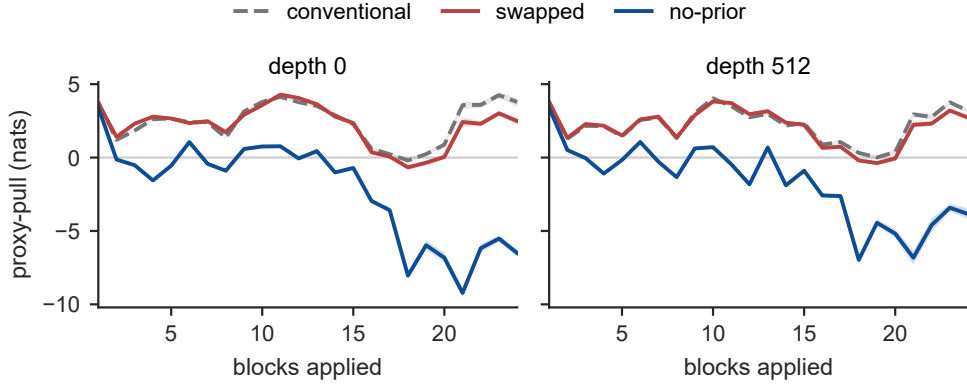


Figure 9: Logit-lens proxy-pull after each block of Qwen2.5-0.5B, as in Figure 3. Swapped follows conventional. No-prior moves toward the binding from blocks 14 to 16 and reaches its lowest point in blocks 18 to 21.

Head	d	import attn		DLA	
		no-pr	swap	no-pr	swap
0.5B L17H1	0	0.79	0.26	-0.27	-0.17
0.5B L17H1	512	0.61	0.09	-0.24	-0.17
1.5B L21H5	0	0.88	0.42	-0.16	+0.03
1.5B L21H5	512	0.86	0.29	-0.15	+0.02
1.5B L22H8	0	0.70	0.23	-0.06	+0.21
1.5B L22H8	512	0.52	0.11	-0.07	+0.21

Table 6: Signature heads at both depths. Attention from the final position to the import line, and direct logit attribution (negative values promote the binding). The contrasts are stable across depths, so they are not a positional effect of the attention sink at the start of the sequence.

tion into a nonce-alias run therefore brings back the convention. The value above 1 is an overshoot that is already present at layer 0. In this direction the import line restores 0.22 to 0.26 of the gap through layer 7 in 0.5B and 0.23 to 0.41 through layer 12 in 1.5B at depth 0, a small direct contribution from the alias embedding on the import line. At depth 512 it stays at or below 0.21. The filler is empty at depth 0 and stays within ± 0.022 at depth 512.

Heads at both depths. Table 6 extends Table 3 to both depths. Over heads whose no-prior import attention exceeds 0.2 at depth 512, mean import attention is 0.56 to 0.59 under no-prior and slightly higher under the swap, so there is no general collapse of attention to the import. The collapse is specific to the heads shown. A separate group of early heads attends to the import more under the swap (1.5B L8H7 goes from 0.04 to 0.90 at depth 512) while contributing almost nothing to the output, consistent with attention that matches repeated tokens (Olsson et al., 2022; Wang et al., 2023).

Model	cond	base	$k=1$	$k=3$
0.5B	swapped	+2.60	+2.52	+2.60
0.5B	conventional	+3.48	+3.45	+3.58
1.5B	swapped	+4.27	+4.74	+4.66
1.5B	conventional	+4.89	+5.43	+5.31

Table 7: Zero-ablation of the top- k heads with positive attribution. Mean proxy-pull against the unablated baseline, 60 items per cell. Results for $k=5$ match $k=3$ within noise (+4.68 and +5.36 for 1.5B, +2.52 and +3.46 for 0.5B). Ablation does not reduce the swapped pull in either model. In 1.5B it raises both conditions, and at $k=1$ it raises the conventional one more (difference +0.07, 95% CI [+0.02, +0.13]).

Ablation. Table 7 gives the ablation results. The 0.5B ranking contains only four heads with positive attribution, so $k=5$ ablates four. Scaling the heads by 0.25 or 0.5 instead of zero gives values between the zero-ablation and baseline values, up to deviations of at most 0.04 nats in 2 of 12 cells.

Mitigation arms. Table 8 gives the mitigation arms with intervals, including the rare pair. We leave out depth 0, where the behavioral pipeline uses a single repetition because repeated prompts would be identical. That leaves two items per cell, too few for a stable interval.

Artifacts. Records, summaries, manifests, and per-experiment figures for this section are in mech/ in the repository. Prompts are stored as hashes, with three examples per condition kept verbatim, and every number reported here can be recomputed from the record files with the included scripts.

Pair	Model	restatement Δ [95% CI]	instruction Δ [95% CI]	no-prior Δ
np $\leftrightarrow$ pd	0.5B	-1.75 [-1.92, -1.57]	-	-9.74
np $\leftrightarrow$ pd	0.5B-inst	-1.53 [-1.74, -1.30]	+0.18 [-0.03, +0.40]	-10.14
np $\leftrightarrow$ pd	1.5B	-2.63 [-2.80, -2.46]	-	-10.53
np $\leftrightarrow$ pd	1.5B-inst	-2.38 [-2.55, -2.20]	-0.85 [-1.08, -0.62]	-10.73
sk $\leftrightarrow$ xgb	0.5B	+0.70 [+0.45, +0.95]	-	-1.50
sk $\leftrightarrow$ xgb	0.5B-inst	+0.50 [+0.21, +0.80]	+2.17 [+1.91, +2.45]	-1.38
sk $\leftrightarrow$ xgb	1.5B	-1.15 [-1.57, -0.74]	-	-3.25
sk $\leftrightarrow$ xgb	1.5B-inst	-1.31 [-1.88, -0.74]	+1.00 [+0.38, +1.58]	-3.60

Table 8: Mitigation arms at depth 512, as differences from the plain swapped baseline (nats, item bootstrap, $n=24$ per cell). The last column is the no-prior condition relative to the same baseline, which is the size of the full gap. The instruction arm is run only on the instruct models. On the strong convention the restatement reliably removes 1.5 to 2.6 nats of a gap of about 10, and the instruction has little effect. On the rare pair the plain baseline is already negative (-1.2 to -3.3), so the models track the binding, and the intervention effects are small and mixed in sign. We draw no conclusion from them.