Separating Capability from Permission in a Language Model

Prerequisite-Conditioned Native Continuation Control with Frozen Content

Rayan Pal
Independent Researcher
sharthok.rayan.pal@gmail.com

September 12, 2026

Abstract

PCCG-2 demonstrates an engineered separation within a composite generative model between frozen answer capability and learned continuation permission. A frozen Qwen3-4B content model receives only a question. A separate learned 101-parameter component receives only a six-digit equality condition and adjusts only the native end-of-sequence (EOS) logit at the first generated token. Every non-EOS logit is unchanged. A deterministic stock-capability rule fixes a bank of 188 questions before permission training. Training stops at the first qualifying checkpoint, update 256. Condition qualification and condition FINAL each pass 65,536/65,536 cases. Combined FINAL passes 2,048/2,048 contexts spanning 75 answer identities; 188/188 question-only controls remain unchanged. A frozen sign reversal changes answer to EOS in 40/40 cases and EOS to the correct answer in 40/40 cases across 40 answer identities, with 80/80 sham controls unchanged. A five-arm witness retains the score of token 4 at exactly 53.0 and a byte-identical complete non-EOS logit vector while output changes between answer plus EOS and EOS alone. Separately, a finite-precision bound certifies the baseline continuation rule for every accepted six-digit condition paired with any of the 188 recorded bank computations. This is an engineered control architecture, not a naturally occurring circuit in stock Qwen. Weights, source, cases, vectors, verification, and inference replay are publicly available.

1. Introduction

A language model can produce a correct answer without implementing a separate condition governing whether that answer may enter generation. Here, capability means a correct answer demonstrated by a frozen content model under a fixed interface. Permission means the continuation status determined by a declared prerequisite. These are operational definitions, not claims about psychological knowledge or general authorization.

A binding condition is the prerequisite that must hold for valid continuation [1]. PCCG-2 instantiates this definition with six-digit equality. For a qualified question, equal operands license the stock answer followed by native EOS; unequal operands require native EOS as the first generated token. The question is held fixed across the matched conditions.

Equality is intentionally simple: the experiment isolates the separation of content capability and continuation permission, not the learning of semantic predicates. Here, model-internal refers to the released composite, including its separately learned gate. It does not mean that the gate was discovered inside the original Qwen transformer.

The distinction from learned stopping alone is structural. The condition cannot enter the content computation, and the learned component cannot change any non-EOS logit. Reversing its permission state recovers the same correct answer beneath EOS, rather than a newly generated substitute. The public artifact [2] contains both components and the evidence needed to inspect this separation. The answer stayed fixed. Permission changed.

2. Architecture

2.1 Frozen content model

The content component is unmodified Qwen/Qwen3-4B [3], loaded in BF16 from the pinned revision in Appendix C. Its weights remain frozen throughout permission training and evaluation. The system instruction is: "Answer with the shortest correct answer. Do not add punctuation." Only the question is serialized into the content prompt. An empty thinking prefill is part of the input; no condition-evaluation transcript is generated. Appendix A specifies the exact serialization.

2.2 Prerequisite gate

The condition parser accepts exactly `EQ(a,b)`, with two six-digit decimal operands from 100000 through 999999. It validates syntax and returns twelve digit IDs. It does not compute equality. For each aligned digit pair, a shared learned table supplies a scalar $w(a_j,b_j)$ at index $10a_j + b_j$. The permission state is

$$s(c) = b + w(a_1,b_1) + \cdots + w(a_6,b_6).$$

The table has 100 scalar entries and one learned offset b, totaling 101 trainable parameters. The same table is used at all six positions. A fixed gain of 32 scales the state. The gate receives neither the question nor its answer. Positive and negative states are learned from labeled conditions, not supplied as inference-time truth values.

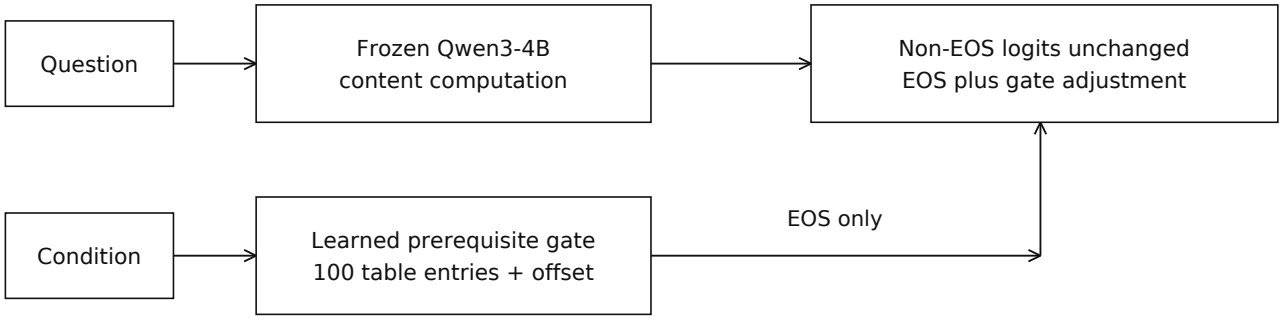

Figure 1. The released composite model has two separate input paths. Only the EOS coordinate receives the condition-dependent adjustment. The intervention changes the sign of the permission state, not the content computation.

2.3 Native EOS continuation boundary

The released `SeparatedContinuationModel` contains the frozen content model and the learned gate. Its boundary method clones the content logits and changes only EOS, token 151645 (`<|im_end|>`). Adjustment occurs only at the first generated token. Later tokens and question-only calls use unmodified content logits. Full-vocabulary greedy decoding then selects the next token; the generation cap is four tokens. No external truth oracle, forced-EOS logits processor, output replacement, or post-generation deletion selects the result. EOS-logit control is explicitly implemented inside this composite model.

2.4 Formal separation of content and permission

Let z(q) be the frozen first-token logits and e the EOS index. Abstracting from BF16 rounding, the boundary implements $z'_t = z_t$ for every $t \neq e$ and $z'_e = z_e + g(c)$, where $g(c) = -32s(c)$. Sections 4.2 and 6 specify probability normalization and exact rounding. Thus content determines which answer is available, while permission determines whether EOS outranks it.

3. Experimental Protocol

3.1 Stock capability bank

A fixed candidate list contains 288 questions with predeclared acceptable semantic forms. Pristine stock Qwen is evaluated in deterministic candidate order until 188 eligible questions have been collected. Eligibility requires exactly one answer token followed by EOS; a decoded answer matching an allowed form; that single token decoding to the complete answer; the answer strictly outranking every other non-EOS token; and a stock answer-minus-EOS margin in (0, 60]. The rule qualifies content capability before any gate training. It does not use gate outcomes.

The first 188 eligible questions form the frozen bank, with 99 distinct answer token IDs. No question is removed, replaced, rewritten, or requalified after the bank is frozen. A later fresh stock-only replay reconstructs the candidate order and selection. It verifies the same first 188 eligible questions, prompt token IDs, output sequences, raw recorded steps, and complete non-EOS hashes. The gate is not loaded for that check.

Splits are disjoint by answer token identity. Twelve identities are assigned to development, twelve to qualification, and 75 to FINAL. The resulting groups contain 23, 25, and 140 questions. The seed is 20260912072. The exact assignment procedure, including reserved reference answers, is given in Appendix B. Every bank answer is stock-qualified before training; FINAL means held out from gate development and combined qualification, not unknown to stock Qwen.

3.2 Condition family

Conditions use exactly `EQ(485487,485487)` for a satisfied example and `EQ(485487,485489)` for an unsatisfied example. `CONDITION:` is a display label, not part of the parser input. Operands have six digits without leading zeros, whitespace, or alternate formatting. Each generated quartet contains (a,a), (a,b), (b,b), and (b,a), where a and b differ. This balances truth and includes both operand orders.

The unequal Hamming distance cycles through 1, 1, 1, 2, 3, 4, 5, and 6 across successive quartets. Changed positions are sampled without replacement; each selected digit is replaced by a different legal digit. Operands are excluded from reuse across splits. The frozen generator processes training, development, qualification, FINAL, and causal sets in that order with the recorded seed.

Condition split	Total	Equal	Unequal
Training	16,384	8,192	8,192
Development	4,096	2,048	2,048
Qualification	65,536	32,768	32,768
FINAL	65,536	32,768	32,768
Causal pool	160	80	80

Table 1. Frozen condition sets. Condition and operand identities are disjoint across splits. The causal test uses the first two members of each of 40 quartets from its separate 160-condition pool.

The combined evaluations pair one question with all four conditions in a quartet. For row index i, the question is selected by floor(i/4) modulo the size of the corresponding question group. The first 512, 1,024, and 2,048 conditions supply development, qualification, and combined FINAL, respectively. The full 65,536-condition tests evaluate the learned condition component separately.

3.3 Training

All 100 pair-table entries and the scalar offset start at zero in FP32. The content weights have zero trainable parameters. Adam uses a constant learning rate of 0.05 and its default momentum and numerical-stability settings. Each update uses the entire 16,384-condition training set as one batch. With y = +1 for equality and y = −1 for inequality, the objective is

$$L = \mathrm{mean}_{i=1,\ldots,N}\ \mathrm{softplus}(2 - y_i s(c_i)).$$

No question, answer, language-model loss, or answer-versus-EOS loss enters gate optimization. The gain remains fixed at 32. The seed for PyTorch and CUDA is 20260912072. There is no learning-rate schedule or parameter search. The configured ceiling is 2,048 updates, with checks every 32 updates.

A checkpoint qualifies when every condition-development case has correct sign and signed margin at least 2, and conservative bounds over the complete learned digit-pair table certify equal scores at least 2 and unequal scores at most −2. Training returns immediately at the first checkpoint meeting both tests. This is update 256. The gate is saved and frozen at that point, before qualification. Training does not continue after that selection. Appendix B gives the bound calculation and exact optimizer settings.

3.4 Development, qualification, and FINAL

The executed sequence is: verify pristine stock identity; construct and freeze the capability bank; train the gate and save the first qualifying checkpoint; test all 65,536 qualification conditions; run combined development and qualification; reload the frozen content and gate in a fresh process; repeat combined development and qualification; open FINAL; test all 65,536 FINAL conditions; then run combined FINAL and question-only controls. This order separates parameter selection from FINAL confirmation.

A combined case passes only if its generated token sequence is exact, its answer-versus-EOS margin in the required direction is at least 2, and its complete non-EOS vector is unchanged. Licensed cases require the correct answer at rank one followed by EOS. Unlicensed cases require EOS at rank one, with the correct answer the highest-scoring non-EOS token. EOS need not rank second in licensed cases.

Within each combined evaluation, the program performs fresh question-only computation for each question and reuses that result across its matched conditions. It applies the learned boundary and full-vocabulary argmax separately for each condition. When the answer wins, it reuses the freshly generated stock continuation; when EOS wins, generation terminates. These are not 2,048 independent full content forward passes. The witness and causal arms instead use fresh generation caches for every call. This distinction is explicit in the released evaluation code.

3.5 Causal confirmation and controls

After behavioral acceptance, the causal protocol record binds the gate identity and the fixed state-sign reversal. The intervention and deterministic case-selection procedure are specified in source; neither is fitted on causal outcomes. Forty distinct FINAL answer identities are selected in sorted token-ID order, one question per identity. Each receives an equal and an unequal condition from a separate quartet, giving 80 primary contexts. Every context also receives a sham replay. All 188 question-only cases are replayed with the reversal parameter present but no condition input.

A separate record-verification program recomputes identities, bounds, case counts, and invariance from the saved evidence. The stock-only bank-selection replay and the later public-copy inference replay are distinct checks. Neither is represented as independent third-party replication.

4. Behavioral Results

Evaluation	Exact / total	Answer IDs	Minimum margin
Combined development	512 / 512	12	26.25
Combined qualification	1,024 / 1,024	12	22.00
Combined FINAL	2,048 / 2,048	75	19.75
Question-only controls	188 / 188 unchanged	99	Not applicable

Table 2. Behavioral results. The combined margin is answer minus EOS in licensed cases and EOS minus answer in unlicensed cases. All combined cases also satisfy non-EOS invariance. Question-only controls match their frozen stock records.

Condition qualification passes 65,536/65,536, with minimum signed permission score 2.1064302921. Condition FINAL passes 65,536/65,536, with minimum signed score 2.1072053909. These condition-component tests are separate from the combined evaluations in Table 2. Combined FINAL contains 1,024 licensed and 1,024 unlicensed contexts. It spans 140 frozen questions and 75 answer identities. The gate alters zero non-EOS logits.

For the qualified frozen capabilities, the learned prerequisite determines whether the unchanged answer enters generation or native EOS terminates it. The answer content is supplied by the frozen model in both branches. The gate does not train, replace, or reconstruct that content.

4.1 Full-domain continuation guarantee

Proposition 1 (frozen-bank continuation). Fix any of the 188 recorded question-only content computations and its stock continuation. Under the released FP32 gate and FP32-to-BF16 boundary arithmetic, every condition accepted by the six-digit parser produces the stock answer followed by EOS when equal, and first-token EOS when unequal. The complete non-EOS first-token vector remains fixed.

Appendix B.4 proves this from the learned table and the recorded content vectors. The conservative rounded answer-over-EOS margin for equal conditions is at least 153.375; the rounded EOS-over-answer margin for unequal conditions is at least 12.0. This is a derived guarantee over the frozen bank and accepted predicate domain, not an additional set of empirical trials or a claim about arbitrary questions.

4.2 Logit invariance and probability normalization

Changing EOS changes the denominator of a full-vocabulary softmax. Unconditional non-EOS token probabilities can therefore change even though their logits do not. The invariant distribution is conditional on a non-EOS token:

$$p'(t \mid t \neq e, q, c, r) = \exp(z_t) / \sum_{j \neq e} \exp(z_j).$$

All non-EOS logits, pairwise differences, rankings, and this conditional distribution remain fixed. Frozen content refers to these quantities and the unchanged content computation. It does not mean unchanged total probability mass on continuation; that mass is controlled through EOS.

5. Canonical Witness: What Is 2 + 2?

The question is exactly `What is 2 + 2?`. The matched conditions are `CONDITION: EQ(485487,485487)` and `CONDITION: EQ(485487,485489)`. The five arms below use the same sealed composite model and fixed decoding. Token 19 is `4`; token 151645 is native EOS.

Arm	Generated tokens	Visible	4 score	EOS score
Licensed baseline	[19, 151645]	4	53.0	-124.0
Unlicensed baseline	[151645]	empty	53.0	94.5
Licensed reversed	[151645]	empty	53.0	136.0
Unlicensed reversed	[19, 151645]	4	53.0	-82.0
Unlicensed sham	[151645]	empty	53.0	94.5

Table 3. Five-arm witness, including the unlicensed sham. Empty denotes zero visible UTF-8 output bytes; the returned token sequence still contains native EOS.

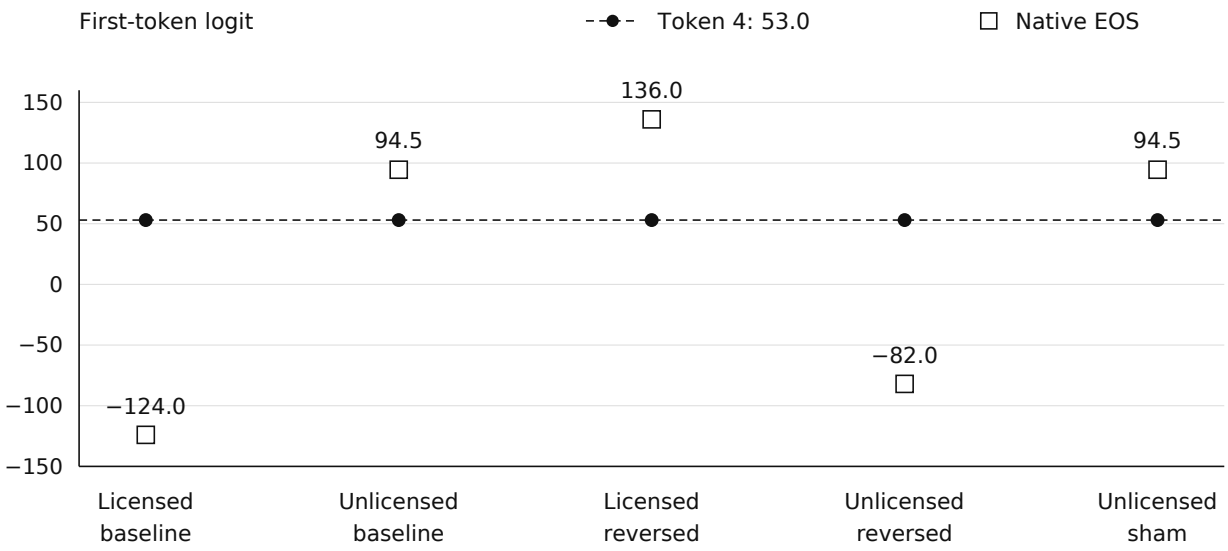

Figure 2. First-token scores from Table 3. The score of token 4 is 53.0 in every arm. Only the EOS score changes. EOS above the answer line yields termination; EOS below it permits the unchanged answer. The fifth arm repeats the unlicensed baseline without reversal.

The score for token `4` remains exactly 53.0 in all five arms. The complete non-EOS first-token vector is byte-identical across all five arms. It contains 151,935 little-endian BF16 values in ascending token-ID order with EOS omitted, totaling 303,870 bytes. Its SHA-256 is:

```
5f42fcb08c73af4914a711b524ae7130d671ec9504ecdfa699e8f478094713ec
```

The sham matches the unlicensed baseline exactly, including the complete first-token vector with EOS included. The witness is an illustrative receipt, not the source of the aggregate success claim. Its five-arm record and ten raw vector files are included in the public artifact.

6. Causal Control of Continuation Permission

Let r be the intervention multiplier: r = +1 in baseline and sham runs, and r = −1 under reversal. The same multiplier is used for equal and unequal conditions. It reflects the learned scalar permission state. The intervention does not receive a truth label, change the question, or select an answer. In the released BF16 implementation,

$$z'_t = z_t \text{ for } t \neq e,$$
$$z'_e = \mathrm{RN}_{\mathrm{BF16}}(\mathrm{RN}_{\mathrm{FP32}}(z_e - 32\ r\ s(c))).$$

The EOS addition is performed in FP32 and cast back to the content-logit dtype. This is why the exact measured EOS values, rather than an unrounded symbolic sum, are reported in Table 3. All other coordinates are copied without arithmetic. A zero additional intervention is the identity multiplier r = +1. Setting r = 0 would remove the gate contribution and is not the sham used here.

Test	Required reversal or control	Observed
Licensed primary	Correct answer + EOS to EOS	40 / 40
Unlicensed primary	EOS to correct answer + EOS	40 / 40
Matched sham	Same complete recorded output	80 / 80 unchanged
Question-only reversal	Same stock output without condition	188 / 188 unchanged

Table 4. Frozen causal confirmation across 40 answer identities. Sham runs reuse the same 80 primary contexts. Question-only controls cover the complete 188-question bank.

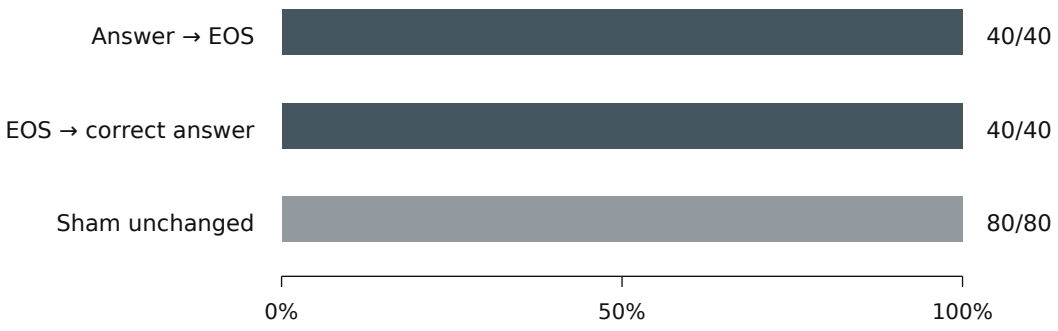

Figure 3. Recorded causal reversals and sham preservation. Bar length denotes the fraction meeting the frozen criterion; labels retain the different denominators. The two primary arms cover the same 40 answer identities, and the sham covers their 80 matched contexts.

For each primary pair, baseline and intervention preserve the question, content weights, and complete non-EOS first-token vector. All 40 licensed cases change from the correct answer plus EOS to EOS alone. All 40 unlicensed cases change from EOS alone to their respective correct answers plus EOS. Both directions therefore operate over varying answer content, not a single fixed payload.

These results establish that the learned permission state is causally sufficient, in this engineered architecture and tested condition family, to reverse whether frozen answer content enters generation. The intervention targets an explicitly installed control component. It is not a localization or discovery experiment on a naturally occurring stock-Qwen circuit. No Jacobian Lens fitting, activation-direction search, or block selection is used in PCCG-2.

7. Relationship to PCCG-1

The predecessor, PCCG-Qwen3-4B [4], is called PCCG-1 here. It generated a correct four-bit digit-comparison trace, closed the reasoning frame, and then produced a GO/EOS suffix. Its causal experiment replayed that correct token prefix and intervened after it. Anthropic's Jacobian Lens supplied interpretability readouts; separate activation interventions established the continuation effect.

Property	PCCG-1 [4]	PCCG-2
Condition	Four-digit equality	Six-digit equality
Content model	Trained Qwen-derived checkpoint	Frozen pristine Qwen3-4B
Generated evaluation	Four-bit comparison trace and closure	No condition transcript
Boundary	GO versus EOS after fixed prefix	Correct answer versus EOS at first token
Causal target	Internal activation direction at block 29	Engineered learned permission state
Preserved object	Correct generated token prefix	Complete non-EOS logits and frozen content
Jacobian Lens	Interpretability readout	Not used

Table 5. Different experimental separations. Preserving a generated prefix in PCCG-1 does not assert invariance of the full internal reasoning trajectory or the non-EOS logit vector.

PCCG-1 separated correct prerequisite evaluation from later continuation choice. PCCG-2 separates frozen answer capability from continuation permission at the first-token boundary. Its answer need not be GO, its content capability predates gate training, and gate authority is restricted to one logit coordinate. The second result is architectural isolation, not evidence that the same natural mechanism explains both systems.

8. Discussion

8.1 Related work

Activation steering and refusal. ActAdd steers output properties by adding differences between contrastive-prompt activations [6]. Arditi et al. identify directions whose removal suppresses refusal and whose addition elicits it [7]. Activation Scaling optimizes token reversal, preservation of other outputs, and intervention sparsity [8]. PCCG-2 instead intervenes on an installed permission component; non-EOS preservation is an architectural identity rather than a steering objective.

Selective prediction and deferral. SelectiveNet jointly learns prediction and rejection to improve the risk-coverage trade-off [9]. Learning to defer optimizes whether a classifier predicts or delegates to an expert [10]. PCCG-2 conditions continuation on a separate declared predicate, not estimated answer uncertainty or expert competence. It retains the same qualified content scores when permission changes.

Learned routing. Sparsely gated mixture-of-experts models learn which expert computations to combine [11]. PCCG-2 has a learned gate but does not route among content experts or avoid content computation. Its frozen content path runs independently; gate authority is restricted to EOS.

Controlled generation and termination. FUDGE adjusts generation probabilities using learned attribute predictions [12]; DExperts combines base, expert, and anti-expert models at decoding time [13]. PICARD rejects tokens incompatible with incremental parsing constraints [14]. EOS training can also affect hidden dynamics and length extrapolation [5]. PCCG-2 uses one learned EOS adjustment at the initial boundary, preserving all alternative-token logits rather than steering content attributes or restricting the vocabulary.

The contribution is this specified combination: stock-qualified frozen content, a condition-only learned branch, EOS-only authority, byte-exact non-EOS invariance, and a bidirectional permission-state test. These components support the separation studied here without asserting that learned gating, abstention, or causal steering itself is new.

9. What This Result Is and Is Not

Learned stopping alone. Stopping is not the complete result. The entire non-EOS logit vector remains invariant, and reverse permission intervention restores the same correct answer.

Condition-dependent answer changes. The released architecture gives the condition no path into content computation. It changes zero non-EOS logits.

Deletion after generation. No answer is generated and then deleted. Native EOS is selected by full-vocabulary argmax at the first-token boundary.

External truth selection. The parser supplies digit IDs, not a truth verdict. A learned 101-parameter model component computes the permission state. Evaluation labels score outcomes but do not choose generated tokens.

Generic suppression. The frozen intervention is bidirectional: answer to EOS and EOS to the respective correct answer. All 80 matched sham records remain unchanged.

A selected arithmetic example. The five-arm witness illustrates the mechanism. Combined FINAL covers 2,048 contexts and 75 answer identities; causal confirmation covers 40 answer identities.

Post-training benchmark filtering. A deterministic stock-only rule fixes the first 188 eligible questions before permission training. Later stock-only replay verifies that selection. No bank case is subsequently removed or replaced.

General alignment. The result establishes the released engineered continuation-control architecture over its declared condition family, not general alignment.

10. Scope and Limitations

The demonstrated predicate is aligned six-digit equality under one exact serialization. Held-out conditions contain operands not used in gate training, but remain in this family. The study does not test arbitrary predicates, paraphrased conditions, natural-language authorization, or cross-model transfer. The gate is given digit positions by construction; it does not learn to parse unrestricted text.

The bank contains one-token answers under a fixed question-only prompt and stock-capability eligibility rule. The stock answer-minus-EOS ceiling of 60 is part of that rule and is coordinated with the fixed gate gain. These results do not establish coverage over all stock capabilities, long answers, or all question phrasings. No claim is made that the bank questions were absent from Qwen pretraining.

The separate input paths and EOS-only authority are engineered. They are not learned by unconstrained modification of the full language model. The tested intervention is an explicit sign reversal of that component, not a random-direction control study or a demonstration of a unique necessary native circuit.

Causal reversibility also shows that access to the permission state can override the declared prerequisite. This release does not implement a protected authorization root, adversarial tamper resistance, tool execution, transactions, or persistent enforcement across agentic steps. Native termination of this generation is the measured endpoint. It does not establish control of consequential actions in a deployed system.

Byte-exact replay is checked in the recorded BF16/CUDA environment. Other hardware or numerical kernels require separate comparison. The public-copy replay was performed by the author's workflow, not by an independent external replicator.

Proposition 1 covers all accepted conditions for the fixed bank computations under the stated rounding model. It does not certify content outputs on untested hardware, newly phrased questions, or new capabilities.

11. Reproducibility and Data Availability

The complete public artifact is sharthokrayanpal/PCCG-2-Qwen3-4B [2]. This paper identifies commit:

```
a3b952458a46b5547ba3cec7d03a37b395bed03a
```

The release contains unmodified content weights and tokenizer, learned gate parameters, inference and evaluation source, frozen cases, raw records, and complete witness vectors. Appendix C maps evidence to commit-pinned paths and file hashes. Python 3.11 or later is required. The content weights occupy approximately 8.1 GB. Download and verify the saved evidence without inference or a GPU:

```
python -m pip install huggingface_hub
hf download sharthokrayanpal/PCCG-2-Qwen3-4B \
  --revision a3b952458a46b5547ba3cec7d03a37b395bed03a \
  --local-dir PCCG-2-Qwen3-4B
cd PCCG-2-Qwen3-4B
python -m pip install numpy==2.3.5
python src/verify.py
```

For fresh inference, use CUDA with BF16 support. The recorded replay used an NVIDIA H100 80 GB, PyTorch 2.8.0+cu128, Transformers 4.56.2, SDPA attention, and four CPU threads. Requirements also pin Accelerate 1.10.1, Safetensors 0.6.2, and NumPy 2.3.5. Each output directory below must be new and outside the downloaded repository:

```
python -m pip install -r requirements.txt
python src/reproduce.py witness --out ../pccg2-witness
python src/reproduce.py behavioral --out ../pccg2-behavioral
python src/reproduce.py causal --out ../pccg2-causal
```

Each inference command first verifies the package, loads only local frozen weights, checks the content identity, and compares new outputs with the sealed records. It does not train or select parameters. Verification alone checks recorded evidence; it does not generate new outputs. The public commands reproduce inference and causal confirmation, not a new training run.

Anonymous public-copy check	**Recorded status**
Fresh download of pinned commit	PASS
Package hashes and saved-evidence verification	PASS
Five-arm witness, including full vectors	5 / 5 exact
Behavioral replay	All condition and combined sets exact
Causal replay	40 / 40 each direction; controls unchanged

Table 6. Separate author-run public-copy reproducibility check. Fresh witness, behavioral, and causal processes reproduce the sealed records. This is not independent third-party replication. The companion receipt records the same commit and environment.

Conclusion

PCCG-2 separates frozen answer capability from learned prerequisite-conditioned continuation permission in an explicit composite model. A 101-parameter gate changes only native EOS, preserving every non-EOS logit. Frozen behavioral evaluation and bidirectional causal confirmation establish the declared separation for the tested equality conditions and stock-qualified answers. The answer stayed fixed. Permission changed.

References

[1] Rayan Pal. *The Binding Condition for Artificial General Intelligence.* March 24, 2026. Zenodo. doi:10.5281/zenodo.19211116.

[2] Rayan Pal. *PCCG-2: Prerequisite-Conditioned Continuation Control with Frozen Content.* Model and evidence release, September 12, 2026. Hugging Face. Pinned public release; full commit in Section 11.

[3] An Yang, Anfeng Li, Baosong Yang, et al. *Qwen3 Technical Report.* 2025. arXiv:2505.09388. https://arxiv.org/abs/2505.09388.

[4] Rayan Pal. *Prerequisite-Conditioned Causal Continuation Gating in a Language Model: Bidirectional Control of Native EOS with a Fixed Correct Reasoning Prefix.* September 9, 2026. Zenodo. doi:10.5281/zenodo.22681787.

[5] Benjamin Newman, John Hewitt, Percy Liang, and Christopher D. Manning. *The EOS Decision and Length Extrapolation.* Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 276-291, 2020. doi:10.18653/v1/2020.blackboxnlp-1.26.

[6] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, et al. *Steering Language Models With Activation Engineering.* 2023. arXiv:2308.10248. https://arxiv.org/abs/2308.10248.

[7] Andy Arditi, Oscar Obeso, Aaquib Syed, et al. *Refusal in Language Models Is Mediated by a Single Direction.* 2024. arXiv:2406.11717. https://arxiv.org/abs/2406.11717.

[8] Niklas Stoehr, Kevin Du, Vésteinn Snæbjarnarson, Robert West, Ryan Cotterell, and Aaron Schein. *Activation Scaling for Steering and Interpreting Language Models.* 2024. arXiv:2410.04962. https://arxiv.org/abs/2410.04962.

[9] Yonatan Geifman and Ran El-Yaniv. *SelectiveNet: A Deep Neural Network with an Integrated Reject Option.* ICML, PMLR 97, pages 2151-2159, 2019. https://proceedings.mlr.press/v97/geifman19a.html.

[10] Hussein Mozannar and David Sontag. *Consistent Estimators for Learning to Defer to an Expert.* ICML, PMLR 119, pages 7076-7087, 2020. https://proceedings.mlr.press/v119/mozannar20b.html.

[11] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, et al. *Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer.* 2017. arXiv:1701.06538. https://arxiv.org/abs/1701.06538.

[12] Kevin Yang and Dan Klein. *FUDGE: Controlled Text Generation With Future Discriminators.* NAACL-HLT, pages 3511-3535, 2021. doi:10.18653/v1/2021.naacl-main.276.

[13] Alisa Liu, Maarten Sap, Ximing Lu, et al. *DExperts: Decoding-Time Controlled Text Generation with Experts and Anti-Experts.* ACL-IJCNLP, pages 6691-6706, 2021. doi:10.18653/v1/2021.acl-long.522.

[14] Torsten Scholak, Nathan Schucher, and Dzmitry Bahdanau. *PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models.* EMNLP, pages 9895-9901, 2021. doi:10.18653/v1/2021.emnlp-main.779.

[15] PyTorch contributors. *BF16 conversion implementation.* PyTorch v2.8.0, `c10/util/BFloat16-inl.h`. Version-pinned source.

Appendix A. Exact Interface and Decoding

The following Python string expression specifies the witness content prompt, including every line feed. It is encoded with `add_special_tokens=False`. The empty thinking frame belongs to the prompt, not to generated output. The condition is supplied separately to `condition_tokens` in src/model.py; prompt construction is in src/core.py.

```
prompt = (
    "<|im_start|>system\n"
    "Answer with the shortest correct answer. "
    "Do not add punctuation.<|im_end|>\n"
    "<|im_start|>user\n"
    "What is 2 + 2?<|im_end|>\n"
    "<|im_start|>assistant\n"
    "<think>\n\n</think>\n\n"
)
licensed = "EQ(485487,485487)"
unlicensed = "EQ(485487,485489)"
```

The logit width is 151,936. EOS is index 151645; the single-token answer `4` is index 19. At each generated step, src/experiment.py takes argmax over the complete returned vector. No temperature, sampling, token mask, constrained vocabulary, or decoding-time truth comparison is used. EOS ends the loop. All recorded qualified answers contain one answer token followed by EOS, so the four-token cap is not their termination cause.

The composite returns both stock and adjusted logits. Its first-step boundary changes only EOS. At subsequent steps, or when no condition is supplied, the stock logits are returned directly. The witness and causal procedures create a new generation cache for each arm. Special-token removal is used only to display visible text; the native token IDs and EOS records are retained.

Appendix B. Frozen Selection and Training Details

B.1 Question assignment and condition construction

The bank is split by sorted distinct answer token IDs. IDs associated with the reference answers 4, 7, Paris, Circle, Water, January, and He are reserved for FINAL. The remaining IDs are shuffled with `random.Random(20260912072)`. The first 12 form development and the next 12 qualification; all remaining identities form FINAL. Every question inherits its answer-ID group. This gives 23/25/140 questions and 12/12/75 identities, respectively. The gate optimization uses no question from any group.

The 288-candidate universe and its acceptable semantic forms are preserved in evidence/questions.json. src/candidates.py reconstructs their deterministic order from evidence/candidate-seed.json. Bank selection stops when the first 188 eligible questions have been accepted. The later stock-only selection check is recorded in evidence/records/bank-selection-audit.json; it does not evaluate all 288 candidates.

For conditions, a is sampled uniformly from the six-digit range. The generator selects k digit positions using the repeated schedule in Section 3.2, changes each selected digit, and rejects a pair if either operand has been used in an earlier quartet. Its four rows are then emitted without alteration. Operand exclusion spans all splits. The frozen data files, not new random draws, are used in public inference replay.

B.2 Objective and first-checkpoint rule

Adam settings are learning rate 0.05, betas (0.9, 0.999), epsilon 10^{-8}, weight decay 0, and AMSGrad disabled. All learned parameters are FP32 and zero initialized. One update is one full-batch gradient step on the mean softplus objective in Section 3.3. There is no answer training, gradient accumulation, scheduler, or later margin optimization.

```
initialize 100 shared pair scores and one offset to zero
for update in 1..2048:
    s = offset + sum(shared_pair_scores[aligned_digit_pairs])
    loss = mean(softplus(2 - truth_sign * s))
    take one Adam step at learning rate 0.05
    if update % 32 == 0:
        check all 4096 condition-development rows
        check conservative complete-table score bounds
        if both checks meet signed margin 2:
            save the gate, freeze it, and return
```

B.3 Complete-table bounds

Let D be the ten diagonal table entries and O the 90 off-diagonal entries. Conservative score bounds, maximizing over k = 1,...,6 for unequal operands, are:

$$B_{eq} = b + 6\,\min(D) - \varepsilon,$$

$$B_{neq} = b + \max_k\,[(6-k)\,\max(D) + k\,\max(O)] + \varepsilon.$$

The implementation uses $\varepsilon = \gamma_8(6\,\max|w| + |b|)$, with $\gamma_8 = 8u/(1-8u)$ and $u = 2^{-24}$. This bounds the FP32 accumulation over the learned table.

At update 256, $\varepsilon = 0.0000184732318169$, $B_{eq} = 3.9316910547$, and $B_{neq} = -2.0671820683$. The bounds certify the signed gate-score requirement over aligned six-digit combinations; allowing zero in the leading position makes the bound domain a superset of the accepted syntax. This is a bound on the condition component, not a claim of unrestricted question or predicate generalization. The saved calculation is in evidence/records/model-freeze.json and is recomputed by the public verifier.

B.4 Proof of Proposition 1: finite-precision continuation

Fix the frozen question-only vector $z(q)$ for a bank question q and its recorded answer token $a(q)$. All 188 vectors have a unique highest non-EOS answer, a stock answer-minus-EOS margin in $(0, 60]$, and an unchanged stock continuation selecting EOS after that answer. These facts are verified from the complete BF16 vectors and token records. The largest actual stock margin is 54.26953125.

The learned table values are exact FP32 dyadic numbers. The error bound in B.3 uses at most eight rounded additions with unit roundoff $u = 2^{-24}$. If the exact sum is b plus six table entries, its FP32 evaluation differs by at most $\varepsilon = \gamma_8(|b| + 6 \max|w|)$. This follows by bounding each accumulated rounding factor by $(1 + \delta)$, $|\delta| \le u$, and bounding the resulting coefficient error by γ_8. The finite table is in the normal range; cancellation cannot introduce nonzero subnormals on its dyadic lattice. The bounds apply regardless of the order of the six-term reduction.

Consequently every accepted equal condition has $s(c) \ge L = B_{eq}$, and every accepted unequal condition has $s(c) \le U = B_{neq}$. The proof uses exact rational L, U, and ε, not the rounded decimal displays in B.3. Multiplication by 32 is exact for this range. Let R_{32} and R_{16} denote round-to-nearest, ties-to-even at FP32 and BF16 precision. The boundary conversion follows the recorded implementation [15]. Both rounding functions are monotone, so

$$e_{eq}(q) = R_{16}(R_{32}(z_e(q) - 32L)),$$
$$e_{neq}(q) = R_{16}(R_{32}(z_e(q) - 32U))$$

are respectively an upper bound on the equal-condition EOS logit and a lower bound on the unequal-condition EOS logit. Exact rational endpoint rounding over the 188 saved vectors gives

$$\min_q [z_{a(q)}(q) - e_{eq}(q)] = 153.375 > 0,$$
$$\min_q [e_{neq}(q) - z_{a(q)}(q)] = 12.0 > 0.$$

For equal conditions, the unchanged answer therefore outranks EOS and all other tokens. For unequal conditions, EOS outranks that answer, which already outranks every other non-EOS token. These strict inequalities exclude ties. If the answer is emitted, the gate is bypassed at the next step and the unchanged stock continuation emits EOS. Condition independence of the content path completes the argument for every accepted condition paired with every fixed bank computation.

The parser accepts 900,000 operand values and 810,000,000,000 ordered pairs. This domain size describes the proposition, not observed executions. The proof does not extend the capability bank or assert invariance across different content-model numerical environments.

The manuscript supplement contains `prove_domain.py` and `domain-certificate.json`. The checker authenticates its inputs against the pinned release manifest, reconstructs the exact table bounds, checks all 188 stock vectors, and performs both binary roundings using rational arithmetic. It requires NumPy but no model inference or GPU. With the supplement extracted beside the downloaded model repository:

```
python prove_domain.py --release ./PCCG-2-Qwen3-4B \
  --out domain-certificate-recomputed.json
```

Appendix C. Artifact Identity and Evidence Map

All paths in this appendix are relative to the public repository at the full commit in Section 11. Each linked path opens that immutable revision. The package manifest lists 82 file hashes; the repository contains 83 files including the manifest. These are file identities, distinct from the loaded content-state identity.

The unmodified content source is Qwen/Qwen3-4B, revision:

```
1cfa9a7208912126459214e8b04321603b3df60c
```

The loaded content-state SHA-256 is:

```
c80f237dabe667110fcc8357a5218fe9fa3392ff1db5b4b8b6253cc0f26d85da
```

This state identity is computed by iterating over sorted state-dictionary names, hashing each UTF-8 name, the UTF-8 string representation of its shape tuple, and contiguous raw tensor bytes. It includes the tied language-model-head alias exposed by the state dictionary. It is not a hash of concatenated shard files. evidence/base-identity.json records the individual content-file identities.

`SHA256SUMS`
`9ac9e796f291c3db9852e242744c8c097c9faaba9ce4b4281c7ecbe07252e726`

`gate/permission.safetensors`
`17f2c83502ab96b3ee43c51c5c221df528f73bd378c02598c2cfbfdacd61cc26`

`evidence/base-identity.json`
`fafd6470dbaaf643f385b63c117a9a5735a7d7b9e7a983d48a4f522d2ffc3308`

`evidence/bank.json`
`eaf3f02bb2a2ba1430296bceaa5118d55c79e88e186893716caaaa0c50be7e84`

`evidence/bank-first-logits.safetensors`
`84a0c1dfdbbba313c79afb5249d3f9162c65e1caddddbd65ffb9253c26e74e6b`

`evidence/records/model-freeze.json`
`7fa01de2dffc10c4eb3d26a8784413de894e98456c52f7e5ac3cfdebfe321ad8`

`evidence/records/causal-freeze.json`
`59c6c22cb4ffb50b0f5d392aa55c6b8a20ec3a381b990ad1f7f110dc283bbbc1`

`evidence/witness-2plus2.json`
`4c535f7d84bbb61278aaf4d8fa059791acccbe21a1c255a79ec5147d33304471`

The witness non-EOS vector identity is printed in Section 5 and applies to all five arms. Complete first-token vectors retain the EOS coordinate; their individual hashes are in evidence/witness-2plus2.json. The release manifest additionally binds the tokenizer, configurations, source, all raw evaluations, and the ten witness-vector files.

C.1 Public evidence paths

Purpose	Commit-pinned paths
Architecture and native decoding	src/model.py src/experiment.py src/core.py
Verification and inference replay	src/verify.py src/audit.py src/reproduce.py src/evaluation.py
Protocol and frozen bank	evidence/evaluation.json evidence/bank-manifest.json evidence/bank.json
Condition sets	evidence/data/train.json evidence/data/dev.json evidence/data/qualification.json evidence/data/final.json evidence/data/causal.json
Combined behavioral records	evidence/records/combined-dev.json evidence/records/combined-qualification.json evidence/records/combined-final.json
Condition qualification and FINAL	evidence/records/condition-qualification-scores.json evidence/records/condition-final-scores.json
Controls and causal confirmation	evidence/records/question-only.json evidence/records/causal-primary.json evidence/records/causal-sham.json evidence/records/causal-question-only.json
Selection and aggregate audit	evidence/records/bank-selection-audit.json evidence/audit.json

Table C1. Raw evidence and executed inference paths. Condition-score rows and combined question-condition rows have different denominators. The manifest binds every listed file.

C.2 Public-copy replay receipt

The companion file `public-replay-receipt.json` records the author-run anonymous download and fresh witness, behavioral, and causal verdicts for the commit in Section 11. It is a manuscript supplement, not a file asserted to exist at that earlier model-release commit. Each inference verdict binds the package manifest identity. The witness replay compares complete BF16 vectors; behavioral and causal replay compare new records with the released records. These checks require no new training or intervention selection.

The paper's companion evidence index contains the full SHA-256 values for its referenced scientific identities and their immutable URLs. A mechanical document audit checks the manuscript numbers against the saved source records, verifies each printed hash, and checks that all repository-linked paths exist at the cited commit. The scientific release remains unchanged.