Can LLMs Reason over Graphs? Formal Expressiveness Bounds and a Hybrid GNN–LLM Framework

Anonymous authors
Paper under double-blind review

Abstract

Large language models (LLMs) are increasingly deployed as general-purpose reasoners over graph-structured inputs, yet their formal expressive power on combinatorial graph problems remains uncharacterised. Empirically, LLMs handle local queries such as vertex degree and short neighbourhood walks reliably, but fail on global properties including graph isomorphism, chromatic number, and long-range connectivity. For message-passing graph neural networks the analogous question is settled, with expressive power tightly characterised by the Weisfeiler–Leman (WL) hierarchy; no comparable bound has so far been established for log-precision transformers reading text-serialised graphs. We close this gap with four results. First, we prove a tight 1-WL upper bound: every log-precision transformer that reads a serialised graph through a permutation-averaged readout has distinguishing power at most 1-WL. Second, we instantiate the bound on two concrete witness families — the Rook$(4, 4)$ vs. Shrikhande pair of strongly regular graphs, and a treewidth-parameterised Cai–Fürer–Immerman ladder. Third, we show that the bound is strictly broken by augmentation: concatenating the transformer with a substructure-counting encoder yields an explicit three-layer in-weights construction that separates the witness pair. Fourth, the same construction extends to bounded-treewidth graph classes, producing a per-MSO_2-formula decidability result whose dependence on the formula length $|\varphi|$ is provably non-elementary. We validate the theory with six experiments and seven ablations across five providers (Llama-3.1-8B, Qwen-2.5-14B, Gemini-2.5-Flash, gpt-4o-mini, gpt-4o): LLM-only accuracy sits at the 0.50 chance line on the Cai–Fürer–Immerman ladder for all five providers and on the strongly regular witness pair for four of five; the hybrid attains perfect accuracy on Gemini-2.5-Flash and gpt-4o via the induced-C_4 feature; and the MSO_2 ladder at treewidth $w = 2$ exhibits the qualitative monotone-ladder shape across providers predicted by the theory. All claims are accompanied by deterministic unit tests in the released code.

1 Introduction

Large language models read graphs as text and answer questions about them. On local queries — vertex degree, neighbour set, a short walk to a named vertex — they perform reasonably well. Push them onto graph isomorphism, chromatic number, or any property that needs a global view of the graph, and they break in a recognisable way (Wang et al., 2023; Fatemi et al., 2024; Tang et al., 2025; Li et al., 2024b; Ge et al., 2025). What is missing from that empirical record is a formal account of *where* the failure boundary lies, and *how* to push past it. The story turns on a Weisfeiler–Leman ceiling: any log-precision transformer that reads a serialised graph through a permutation-averaged readout sits at most at 1-WL. Pair the transformer with a k-WL-equivalent GNN encoder, and that ceiling lifts at the cost of $\mathcal{O}(n^{k+1})$ extra compute per round on the encoder side.

1.1 An Empirical Failure Without a Formal Explanation

The empirical picture. The shape of the failure is by now well documented (Wang et al., 2023; Fatemi et al., 2024; Li et al., 2024b; Tang et al., 2025; Ge et al., 2025); the *why* is not. For message-passing GNNs the situation is sharper. Morris et al. (2019), Xu et al. (2019), and Maron et al. (2019) ladder the expressiveness directly to k-WL.

But no such ladder exists for log-precision transformers reading text-serialised graphs. The closest entries either model attention abstractly (Joshi, 2025; Sanford et al., 2024b;a) or characterise the language classes accepted by the architecture (Merrill & Sabharwal, 2023; Chiang et al., 2023; Merrill & Sabharwal, 2025; Barceló et al., 2024) — neither produces a graph-side WL upper bound. That is the gap this paper closes.[1]

> *What, formally, is the expressive ceiling of a log-precision transformer that reads a graph through a fixed serialisation, and how do we break through it?*

1.2 Contributions

Our contributions. Four results, summarised here and stated formally in Section 5.

1. **The 1-WL ceiling (Theorem 5.1).** Every log-precision transformer paired with a permutation-invariantised readout has distinguishing power at most 1-WL. The proof reduces the transformer to a multiset-respecting TC^0 circuit family (Merrill & Sabharwal, 2023), then to 1-WL refinement (Morris et al., 2019; Xu et al., 2019), with the residual ordering dependence eliminated by averaging over $\pi \sim \mathrm{Uniform}(\mathrm{S}_n)$.
2. **Concrete witnesses (Proposition 5.3).** Two families of 1-WL-equivalent yet non-isomorphic pairs: $\mathrm{Rook}(4,4)$ vs. Shrikhande on 16 vertices for the small-instance case, and the Cai–Fürer–Immerman ladder (Cai et al., 1992) parameterised by treewidth for scale.
3. **Strict hybrid superiority (Theorem 5.4).** A 3-layer transformer fed the digest of a substructure-counting encoder Φ that emits $\#K_4(G)$ — instantiable as K_4-aware GSN (Bouritsas et al., 2023), 3-FWL refinement (equivalent to 4-WL by the k-FWL $\equiv (k+1)$-WL identity (Maron et al., 2019)), or direct $\mathcal{O}(n^4)$ enumeration — answers "does G contain a K_4?" on the SRG witness, strictly beyond the reach of either component alone (and beyond 2-FWL $\equiv$ 3-WL, which is itself 1-WL-collapsed on this pair (Bouritsas et al., 2023)). We give the explicit in-weights program.
4. **Treewidth-parameterised hierarchy (Theorem 5.5).** For every MSO_2 sentence φ and width bound w, there exists a hybrid that decides φ on $\mathcal{C}_w$, with the precision p_0 needed to encode the lookup-table biases (and the offline construction time for those biases) depending non-elementarily on $|\varphi|$ following Courcelle (1990); Seese (1991); Frick & Grohe (2004).

Empirical validation. Six experiments and seven ablations against five LLM providers — Llama-3.1-8B-Instruct, Qwen-2.5-14B-Instruct (both open-source, 4-bit quantised via Ollama), Gemini-2.5-Flash, gpt-4o-mini, gpt-4o (closed-source via `litellm`). LLM-only at the 0.50 chance line on CFI for all five providers and on SRG for four of five; Llama-3.1-8B's 1.00 on SRG is isolated as memorisation of the textbook $\mathrm{Rook}(4,4)$/Shrikhande pair via four control cells (full discussion in Section H.8). The hybrid hits 1.00 on Gemini Flash and gpt-4o via the induced-C_4 feature; the MSO_2 ladder at $w = 2$ produces a qualitative monotone-ladder shape on the 3-colourability column across providers in capability order, the predicted picture-form of Theorem 5.5(c) (full per-cell numerics and the Bonferroni honesty clause in Section 7). Every finding lines up with one of the four theorems and is pinned by a test in `src/theory/`. Figure 1 is the architecture-WL map.

Paper organisation. Sections 2 to 4 cover related work, notation (WL, MSO, log-precision transformer), and the four research questions with f_T as the unit of analysis. Section 5 states the four theorems; Section 6 describes the constructive hybrid; Section 7 runs the empirical battery; Section 8 closes with limitations and open problems. Full proofs and supplementary experiments are in Sections B to I.

[1] For scale: the experimental matrix in Section 7 runs in 371 s wall-clock for Gemini 2.5 Flash on a single 2024 Apple-Silicon laptop against Tier 1 quota (1,000 RPM, 10,000 RPD), under USD 1 for the OpenAI providers combined, etc. — Section G.3 has the per-experiment breakdown.

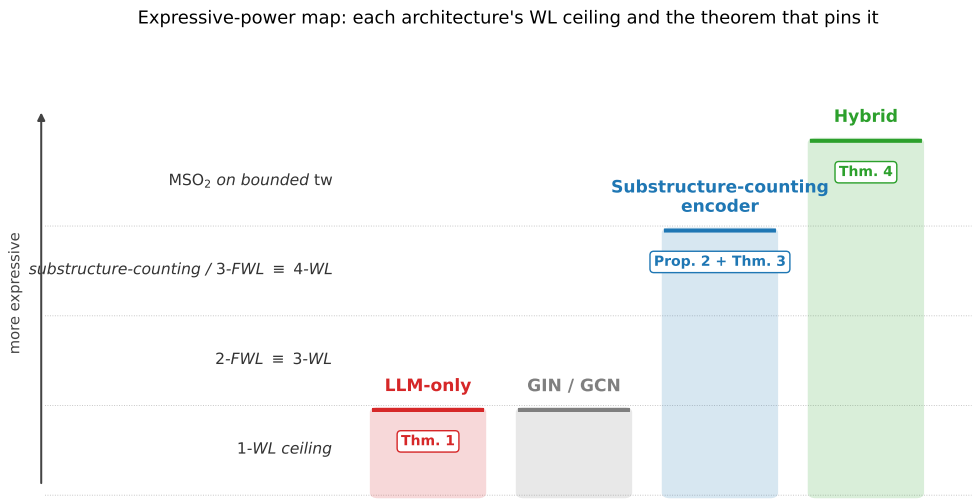


Figure 1: Architecture-vs-WL ladder and the theorems that pin each level. LLM-only and standard message-passing GNNs share the 1-WL ceiling (Theorem 5.1); the witnesses of Proposition 5.3 sit just above. A k-FWL / substructure-counting encoder lifts the ceiling (Theorem 5.4); on bounded-treewidth classes the same lift yields per-MSO$_2$-formula decidability (Theorem 5.5).

2 Related Work

This paper sits at the intersection of four threads: log-precision transformer expressiveness, GNN expressiveness via the Weisfeiler–Leman hierarchy, empirical LLM benchmarks for graph reasoning, and the treewidth–MSO bridge.

Transformer expressiveness. The literature brackets two unbounded regimes: universal approximation (Yun et al., 2020) and Turing-completeness (Pérez et al., 2021) both assume conditions vacuous at log precision. The bounded regime is sharpened by Merrill & Sabharwal (2023) (log-precision ⇒ logspace-uniform TC^0) and Chiang et al. (2023); Barceló et al. (2024) characterise hard-attention encoders via counting logic. For graph-structured inputs: Joshi (2025) formalises attention-as-MPNN; Sanford et al. (2024b;a) separate graph-algorithmic tasks by transformer depth/width; Merrill & Sabharwal (2025) shows log-depth (rather than constant-depth) suffices for connectivity. Theorem 5.1 synthesises these into a graph-side WL-style ceiling.

GNN expressiveness and the WL hierarchy. Morris et al. (2019) and Xu et al. (2019) established the 1-WL ceiling for message-passing GNNs and the GIN saturating it; Maron et al. (2019) extended to k-WL via k-FWL-equivalent architectures, which we use as the hybrid encoder. Müller et al. (2024); Müller & Morris (2024) bring the same expressiveness to graph transformers via k-WL-aligned tokenisation. Cai et al. (1992) give the canonical hard pairs (CFI ladder, used in Proposition 5.3(b)); the matching positive direction is Kiefer & Neuen (2022): w-WL is complete on $\mathcal{C}_w$, used by Theorem 5.5(b). For structurally-restricted classes we cite Kiefer & Neuen (2019); Grohe & Kiefer (2019); Kiefer et al. (2019) (planar / bounded-genus) and Levet et al. (2024) (parallel WL canonisation on bounded rank-width). Dell et al. (2018) characterise WL equivalence as hom-count equivalence over bounded-treewidth patterns — the dual treewidth statement complementing our input-side use of $\mathcal{C}_w$.

LLMs for graph reasoning. Empirical work has mapped *where* LLM graph reasoning fails but not *why*. NLGraph (Wang et al., 2023), Talk-Like-a-Graph (Fatemi et al., 2024), GraphArena (Tang et al., 2025), and GLBench (Li et al., 2024b) report aggregate accuracy on natural-language graph tasks; Ge et al. (2025) show that LLM accuracy is not permutation-invariant, which complements our permutation-averaging story directly. Li et al. (2024a) situates LLMs as code-writers rather than direct reasoners. The 2024–2025 wave of GNN–LLM hybrids includes GraphGPT (Tang et al., 2024), LLaGA (Chen et al., 2024b), GraphLLM (Chai et al., 2023), InstructGLM (Ye et al., 2024), GraphToken (Perozzi et al., 2024), and GraphWiz (Chen et al., 2024a); Mao et al. (2024) crystallises the position. These are engineering recipes that do not specify

when the hybrid succeeds where its parts fail; Theorem 5.4 fills that gap with an explicit witness pair, and Theorem 5.5(c) generalises to an MSO_2 family. We do not run head-to-head against these hybrids in the present paper because the unit of analysis is different: their benchmarks report aggregate accuracy on natural-language graph tasks where the underlying graph properties span many WL levels and the dominant failure mode is prompt brittleness, while ours is per-WL-level performance on adversarial counterexamples designed so that a w-WL-bounded reader provably cannot distinguish the pair, regardless of prompt. The natural follow-up protocol — port their published prompt templates to our srg(16, 6, 2, 2) and CFI-ladder inputs at matched LLM and encoder budgets and report per-cell accuracy alongside our Table 2 — is the cleanest way to extend our WL-level diagnosis to those hybrid families, and is the most load-bearing follow-up (Section 8).

Treewidth, MSO, and algorithmic meta-theorems. Courcelle (1990): every MSO_2 sentence is decidable in $f(\mathrm{tw}, |\varphi|) \cdot n$ on bounded-treewidth graphs. Seese (1991): converse (decidable MSO_2-satisfiability implies bounded tw). Frick & Grohe (2004): non-elementary f. These three results bracket the parameter exactly and are what Theorem 5.5(c) inherits. Linear-time treewidth approximation: Bodlaender (1996). Quasipolynomial graph isomorphism: Babai (2016). To our knowledge Theorem 5.5 is the first application of this structural-graph-theory stack to the LLM–GNN–hybrid hierarchy.

3 Background and Notation

All graphs are simple, undirected, $|V| = n$, $|E| = m$. Table 3 in Section A collects every symbol used; each definition has a unit test in `tests/`.

3.1 The Weisfeiler–Leman Hierarchy

Convention. We use the GNN-literature convention: 1-WL is colour refinement on vertices, and k-WL for $k \geq 2$ refines on ordered k-tuples (Cai et al., 1992). On simple unlabelled graphs 1-WL $\equiv$ 2-WL (the standard collapse), so the strict hierarchy is 1-WL $\equiv$ 2-WL $<$ 3-WL $< \cdots$, and k-FWL of Maron et al. (2019) satisfies k-FWL $\equiv (k+1)$-WL.

Definition 3.1 (1-WL refinement)**.** For $G = (V, E)$, let $c_G^{(0)}(v) = \mathsf{c}_0$ for a constant c_0, and define $c_G^{(t+1)}(v) = \mathrm{HASH}\big(c_G^{(t)}(v), \{\{c_G^{(t)}(u) : u \in N(v)\}\}\big)$. We write $G_1 \equiv_{1\text{-WL}} G_2$ when the stable colour histograms coincide. Cai et al. (1992) construct graph pairs for which no fixed k-WL distinguishes them.

Definition 3.2 (k-WL refinement)**.** For $k \geq 2$, k-WL colours ordered k-tuples $\bar{v} \in V^k$; the initial colour is the isomorphism type of the induced ordered subgraph on $\bar{v}$, and the update replaces the j-th coordinate by every $w \in V$ in turn and aggregates the resulting multiset of refined colours. We write $G_1 \equiv_{k\text{-WL}} G_2$ when the stable k-tuple histograms coincide; $\dim_{\mathrm{WL}}(G)$ is the minimal k at which k-WL identifies G. The full k-WL update rule appears in Section A.

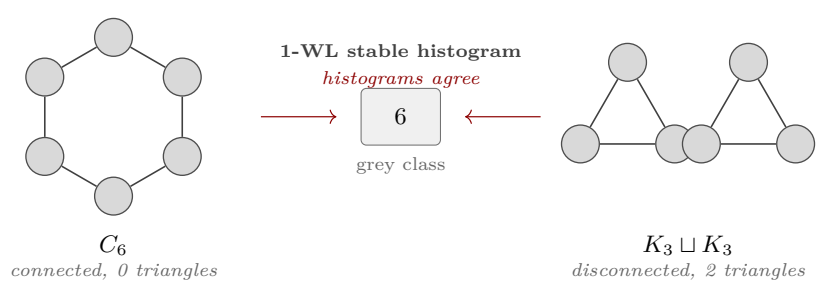


Figure 2: 1-WL on the smallest non-trivial counterexample. Both C_6 and $K_3 \sqcup K_3$ are 2-regular on 6 vertices; 1-WL stabilises after one round with all six vertices in a single class and identical histograms (centre). But C_6 is connected and triangle-free, while $K_3 \sqcup K_3$ is disconnected with 2 triangles — either of these is computable by 2-FWL ($\equiv$ 3-WL). This is the smallest example of the gap Theorem 5.1 pins as an upper bound on f_T.

3.2 Message-Passing Neural Networks

Xu et al. (2019) prove that with injective aggregation, an MPNN matches 1-WL exactly as a graph distinguisher: $h_v^{(\ell+1)} = \mathrm{UPD}^{(\ell)}\big(h_v^{(\ell)}, \mathrm{AGG}^{(\ell)}(\{\{h_u^{(\ell)} : u \in N(v)\}\})\big)$ with a multiset-respecting readout.

3.3 Log-Precision Transformer Encoders

Definition 3.3 (Log-precision transformer encoder)**.** $T_{L,H,d,p}$ has L layers, H heads, embedding dimension d, and per-activation precision p bits. Multi-head softmax attention is followed by an FFN with residual + LayerNorm; intermediate values are rounded to p bits. We fix $p = c \log n$ throughout, the regime in which a constant-depth transformer computes a function in logspace-uniform TC^0 (Merrill & Sabharwal, 2023). Motivation for this regime (bf16, deployed-model token counts) and historical context vs. Pérez et al. (2021); Yun et al. (2020) appear in Section A.

3.4 Graph Serialisation

A serialisation is a function $\mathsf{S} : \mathcal{G} \times \mathrm{S}_n \to \Sigma^*$ taking a graph and a node ordering to a token string. We use three canonical formats (implemented in `src/serialization/`): an adjacency-list format that lists each vertex's π-sorted neighbours, an edge-list format that emits one ordered pair per edge, and a natural-language format that wraps the same data in an English template. A single forward pass $T(\mathsf{S}(G, \pi))$ depends on π, so the natural object of analysis is the permutation-invariantised readout $f_T(G) = \mathbb{E}_\pi[\rho(T(\mathsf{S}(G, \pi)))]$ defined in Section 4.1; at test time we Monte-Carlo it with $K = 4$ permutations.

3.5 Treewidth and Monadic Second-Order Logic

Definition 3.4 (Treewidth)**.** A tree decomposition $(T, \{B_t\}_{t \in V(T)})$ of G satisfies $\bigcup_t B_t = V$; every edge appears in some bag; and for every v the bag-set containing v is a connected subtree. The width is $\max_t |B_t| - 1$; the treewidth $\mathrm{tw}(G)$ is the minimum over all decompositions. Examples: $\mathrm{tw}(\text{tree}) = 1$, $\mathrm{tw}(C_n) = 2$, $\mathrm{tw}(K_n) = n - 1$, $\mathrm{tw}(k \times k\text{-grid}) = k$. The class $\mathcal{C}_w = \{G : \mathrm{tw}(G) \le w\}$ is minor-closed; Bodlaender (1996) decides $G \in \mathcal{C}_w$ in $\mathcal{O}(c_w \cdot n)$; Kiefer & Neuen (2022) prove $\dim_{\mathrm{WL}}(\mathcal{C}_w) \le w$ for $w \ge 2$ (the operative identity for Theorem 5.5(b)).

MSO_1 / MSO_2. MSO_1 extends first-order logic on graphs with quantification over vertex sets; MSO_2 additionally quantifies over edge sets. Courcelle (1990) proves every MSO_2 sentence φ is decidable in $f(\mathrm{tw}, |\varphi|) \cdot n$ time on bounded-treewidth graphs, with f non-elementary in $|\varphi|$ (Frick & Grohe, 2004); the converse (decidable MSO_2-satisfiability implies bounded treewidth) is Seese (1991). The same non-elementary f reappears as the bit-precision p_0 in Theorem 5.5(c). Worked examples (3-colourability in MSO_1, Hamiltonicity in MSO_2) appear in Section A; the homomorphism-count characterisation of WL equivalence (Dvořák, 2010; Dell et al., 2018) also lives there.

4 Problem Setup

4.1 Permutation-Invariantised Readout

A single forward pass $T(\mathsf{S}(G, \pi))$ depends on the tie-breaks of π in a way that is uninteresting graph-theoretically. We therefore analyse the *permutation-invariantised readout*

$$f_T(G) \;:=\; \mathbb{E}_{\pi \sim \mathrm{Uniform}(\mathrm{S}_n)}\big[\rho\big(T(\mathsf{S}(G, \pi))\big)\big],$$

where $T = T_{L,H,d,p}$ with $p = c \log n$ (Merrill & Sabharwal, 2023) and (L, H, d) arbitrary constants in n, and $\rho : \mathbb{R}^{n \times d} \to Y$ is a bounded readout. $f_T : \mathcal{G} \to Y$ is well-defined and is the right object to compare across graphs. At test time we Monte-Carlo it with $K = 4$ permutations (mean-squared error $\mathcal{O}(K^{-1/2})$); Section H.2 discusses equivalent constructions (canonical-ordering majority vote, $(\min, \max, \mathrm{mean}, \mathrm{median})$ tuple), the K sweep, and motivation for working with f_T rather than raw T.

4.2 Research Questions

Question 4.1 (WL ceiling, answered by Theorem 5.1)**.** What is the maximum WL level that f_T can simulate on a serialised graph?

Question 4.2 (Hybrid gain, answered by Theorem 5.4)**.** Does concatenating a k-WL-equivalent GNN encoder's output to the prompt provably exceed the 1-WL ceiling?

Question 4.3 (Treewidth parameterisation, answered by Theorem 5.5)**.** How does $\mathrm{tw}(G)$ modulate the expressiveness gap between LLMs, GNNs, and hybrids?

Question 4.4 (Empirical alignment, answered by Section 7)**.** Do real LLMs fail where the theory predicts (the WL boundary and the treewidth hierarchy)?

5 Main Results

Four headline results: a 1-WL ceiling for f_T (Theorem 5.1); explicit witness families on which the ceiling binds (Proposition 5.3); a strict hybrid separation on one fixed witness pair (Theorem 5.4); and a treewidth-parameterised hierarchy locating the LLM, GNN, and hybrid relative to the MSO_2 decidability boundary (Theorem 5.5). Full proofs in Sections B to E.

5.1 The 1-WL Ceiling

Theorem 5.1 (1-WL upper bound; graph-side corollary of Merrill & Sabharwal, 2023 via S_n orbit-averaging)**.** *Let $T = T_{L,H,d,p}$ be a log-precision transformer with $p = c \log n$ for a constant $c > 0$ and fixed (L, H, d). Let S be any of the three serialisations of Section 3.4 and let $\rho : \mathbb{R}^{n\times d} \to Y$ be any bounded readout into a finite codomain Y. Then for all graphs G_1, G_2 on n vertices with $G_1 \equiv_{1\text{-WL}} G_2$,*

$$f_T(G_1) \;=\; f_T(G_2) \quad \textit{whenever} \quad L \cdot 2^{-p} \;<\; \tfrac{1}{2} \min_{y \neq y' \in Y} |y - y'|.$$

For real-valued readouts the same equality holds up to an additive rounding error of size $\mathcal{O}(L{\cdot}2^{-p})$. In particular, T cannot decide isomorphism on any 1-WL-equivalent pair of non-isomorphic graphs once $p = c \log n$ is large enough to keep the rounding error below the readout's quantisation step.

Proof sketch (Section B). Four lemmas: (A) attention-as-MPNN on the complete bipartite token graph (Lemma B.1, in TC^0 (Merrill & Sabharwal, 2023)); (B) adjacency-list serialisation factors per-vertex blocks through local 1-WL data (Lemma B.2); (C) a bit-exact bijection-invariance induction on layers (Lemma B.3) yields per-position factorisation $x^{(\ell)}(i) = \Psi(\mathrm{role}(i; G, \pi), \mathrm{LP}(G, \pi))$; (D) S_n-averaging the readout collapses the labelled-partition orbit to the WL histogram (Lemma B.5). □

Remark 5.2 *(What Theorem 5.1 does not say).* The bound is for a single-pass log-precision transformer with permutation-averaged readout. It does not rule out chain-of-thought (a quantitative CoT-length-vs-WL-level trade-off is an open question; see Section F for a proof outline tying Merrill & Sabharwal, 2024; Li et al., 2024c to k-WL via the CFI families), tool-use such as the hybrid of Theorem 5.4, or the unbounded-depth/precision regime of Yun et al. (2020); Pérez et al. (2021). The depth-bounded form $\equiv_{L\text{-iterWL}}$ also holds when $L < \mathrm{diam}(G)$ (Section B.3).

5.2 Explicit Indistinguishable Pairs

Proposition 5.3 (Concrete 1-WL-equivalent witnesses)**.** *Two families of non-isomorphic 1-WL-equivalent pairs:*

(a) ***SRG witness.*** $\mathrm{Rook}(4, 4)$ *and* $\mathrm{Shrikhande}$*: both* $\mathrm{srg}(16, 6, 2, 2)$*, with* $|\mathrm{Aut}(\mathrm{Rook}(4, 4))| = 1152$ *and* $|\mathrm{Aut}(\mathrm{Shrikhande})| = 192$*.*

(b) ***CFI ladder.*** *For every $k \geq 1$, taking a base graph X with $\mathrm{tw}(X) = k + 1$ (e.g. $X = K_{k+2}$) yields a non-isomorphic pair $(\mathrm{CFI}(X), \mathrm{CFI}'(X))$ that is k-WL-equivalent (Cai et al., 1992; Kiefer & Neuen, 2022).*

By Theorem 5.1, no log-precision transformer with fixed (L, H, d) and permutation-averaged readout solves isomorphism on these witnesses.

Proof sketch (Section C). (a) Any two $\mathrm{srg}(n, k, \lambda, \mu)$ graphs are 1-WL-equivalent because 1-WL on a regular graph with uniform initial colouring stabilises trivially (Lemma C.1); non-isomorphism follows from the automorphism-count asymmetry (or equivalently, induced-C_4 counts 36 vs. 12). (b) The Cai–Fürer–Immerman construction (Cai et al., 1992) via the $\dim_{\mathrm{WL}}(\mathrm{CFI}(X)) = \mathrm{tw}(X)$ identity of Kiefer & Neuen (2022) (Lemma C.3). □

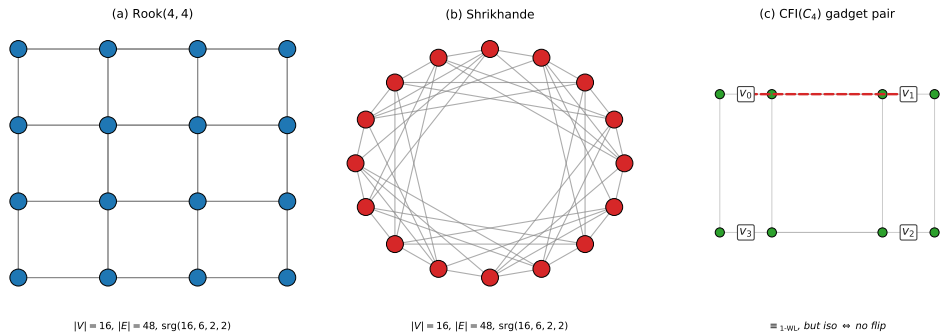


Figure 3: Witnesses of Proposition 5.3. **(a)** Rook$(4, 4)$ on its 4×4 grid; **(b)** Shrikhande in circular layout — both $\mathrm{srg}(16, 6, 2, 2)$, hence 1-WL-equivalent yet non-isomorphic. **(c)** A $\mathrm{CFI}(C_4)$ gadget pair; the dashed red edge marks the parity flip distinguishing $\mathrm{CFI}(C_4)$ from $\mathrm{CFI}'(C_4)$.

5.3 Strict Hybrid Superiority

Theorem 5.4 (Strict hybrid separation)**.** *Let Q_* be the Boolean question* "does G contain a K_4 clique?"*. There exists a hybrid $H = (T_3, \Phi)$ consisting of a 3-layer log-precision transformer T_3 and a graph encoder Φ that outputs $\#K_4(G)$ as a graph-level scalar (concretely instantiated by K_4-aware GSN (Bouritsas et al., 2023), by 3-FWL refinement (Cai et al., 1992; Maron et al., 2019), or by direct $\mathcal{O}(n^4)$ subgraph enumeration), such that:*

(a) H solves Q_* on $(\mathrm{Rook}(4, 4), \mathrm{Shrikhande})$ *exactly;*
(b) No log-precision transformer of bounded size, with any of the three serialisations and any permutation-averaged readout, solves Q_* on this pair (consequence of Theorem 5.1).

The construction is explicit (Section 6; full proof Section D).

Proof sketch (Section D). (a) Explicit in-weights program: Stage 1 encoder Φ produces $\#K_4(G)$ ($= 8$ for $\mathrm{Rook}(4, 4)$, $= 0$ for Shrikhande); Stage 2 transformer with three heads (question-parser, digest-copy, threshold) emits the Boolean. (b) Direct from Theorem 5.1. The classical 2-FWL $\equiv$ 3-WL collapse on $\mathrm{srg}(16, 6, 2, 2)$ pairs (Cai et al., 1992; Bouritsas et al., 2023) explains why a substructure-counting encoder (rather than a 3-WL one) is the right architectural fix. □

5.4 Treewidth-Parameterised Hierarchy

Theorem 5.5 (Treewidth-parameterised hierarchy)**.** *Let $\mathcal{C}_w = \{G : \mathrm{tw}(G) \leq w\}$. Then:*

(a) ***LLM on $\mathcal{C}_w$.*** *f_T has distinguishing power at most 1-WL on $\mathcal{C}_w$. For $w = 1$ (forests) 1-WL is complete; for $w \geq 2$, Proposition 5.3(b) supplies non-isomorphic 1-WL-equivalent pairs in $\mathcal{C}_w$.*
(b) ***GNN on $\mathcal{C}_w$.*** *A w-WL-equivalent GNN identifies every graph in $\mathcal{C}_w$ (Kiefer & Neuen, 2022; Morris et al., 2019).*
(c) ***Hybrid on $\mathcal{C}_w$ (per-formula existence, non-uniform in φ).*** *For every MSO_2 sentence φ, every $w \in \mathbb{N}$, and every $n_{\max} \in \mathbb{N}$, there exist a constant L_0 (independent of φ) and constants $d_0(\varphi, w, n_{\max})$,*

$p_0(\varphi, w, n_{\max})$ such that the hybrid H_{L_0,d_0,p_0} decides φ on every $G \in \mathcal{C}_w$ with $|V(G)| \leq n_{\max}$. The construction is non-uniform in φ — a different hybrid for each formula; we do not claim a single hybrid handling all MSO_2 *sentences. Bit-precision p_0 depends non-elementarily on $|\varphi|$, matching the Courcelle–Seese blow-up (Courcelle, 1990; Seese, 1991; Frick & Grohe, 2004).*

(d) ***Hardness ladder.*** *For every $k \geq 1$, the CFI construction on a base graph X with $\mathrm{tw}(X) = k+1$ yields a non-isomorphic k-WL-equivalent pair in $\mathcal{C}_{O(k)}$, indistinguishable by any system whose effective WL level is at most k.*

Proof sketch (Section E). (a) Theorem 5.1 restricted to $\mathcal{C}_w$, with boundary cases $w = 1$ and $w \geq 2$. (b) Compose Kiefer & Neuen (2022) ($\dim_{\mathrm{WL}}(\mathcal{C}_w) \leq w$ for $w \geq 2$) with the w-dim GNN $\equiv$ w-WL identity. (c) GNN identification plus Courcelle's algorithm packaged as a TC^0 lookup over the finite GNN-embedding image, embedded in a constant-depth log-precision transformer via Merrill & Sabharwal (2023). (d) Lemma C.3 applied at $\mathrm{tw}(X) = k+1$. □

Remark 5.6 *(Parameter regime for* 5.5*(c)).* Depth L_0 is a small absolute constant, width d_0 grows polynomially in $|\varphi|$ at fixed $n_{\max}$; only the bit-precision p_0 carries the Courcelle–Seese non-elementary blow-up.

6 The Hybrid Architecture

Theorem 5.4 proves the existence of a hybrid lifting the 1-WL ceiling; this section describes the concrete object we built and ran. Recipe: run a k-WL-equivalent GNN encoder, serialise G in one of the three formats of Section 3.4, concatenate the encoder digest with the serialisation and the question, hand the resulting prompt to a black-box transformer. The pipeline is sketched in Figure 4 and given as pseudocode in Algorithm 1; vector- vs. token-injection trade-offs and computational-cost breakdown are in Section D.5. The empirical instantiation lives in `src/models/hybrid.py`; the constructive Theorem-3 witness in `src/theory/hybrid_witness.py` hard-codes the in-weights program for the K_4 question.

6.1 Design Principles

Match the encoder's WL dimension to the input class. By Theorem 5.5(b), a w-WL-equivalent encoder is a complete invariant on $\mathcal{C}_w$. Pushing w above the test treewidth is wasted compute; below leaves witness pairs the encoder cannot distinguish.

Numeric prefix injection. The encoder's digest enters the prompt as a comma-separated list of fixed-precision p-bit numeric values, preserving the $\mathcal{O}(\log n)$-bit content per coordinate and avoiding string-float rounding noise.

LLM as black box. We treat the transformer as black-box, accessed through the OpenAI-compatible `litellm` interface — no fine-tuning, no soft prompts, no retrieval.

6.2 Treewidth-Aware Variant (Bounded-tw Inputs; Fixed-k Fallback Elsewhere)

An optional pre-processing step estimates $\mathrm{tw}(G)$ via the 2-approximation of Bodlaender (1996) and picks the encoder's WL dimension to match: if $\widehat{\mathrm{tw}} = w$, we use a w-FWL encoder. On bounded-treewidth inputs the adaptation cuts compute by skipping unnecessary high-order WL refinements; on the unbounded-treewidth regime that most real-world graphs (knowledge graphs, social networks) inhabit, the variant degrades to the fixed-k hybrid and offers no asymptotic saving. Fixed-k curves appear in Section H.12; the adaptive variant in Section H.1. The adaptive gate routes to the encoder only when LLM confidence $|s - 0.5| < \tau$ for $\tau \in \{0.10, 0.25\}$, both of which retain the always-on hybrid's 0.80 accuracy on the MSO sub-ladder.

7 Empirical Evaluation

Six experiments (E-1..E-6) and seven ablations (A-1..A-3, P-2..P-5) test the four theorems of Section 5. Mapping experiments to theorems: E-1 instantiates Proposition 5.3; E-2 stresses the serialisation invariance of Theorem 5.1; E-3 and E-4 trace the treewidth- and formula-complexity boundaries of Theorem 5.5; E-5

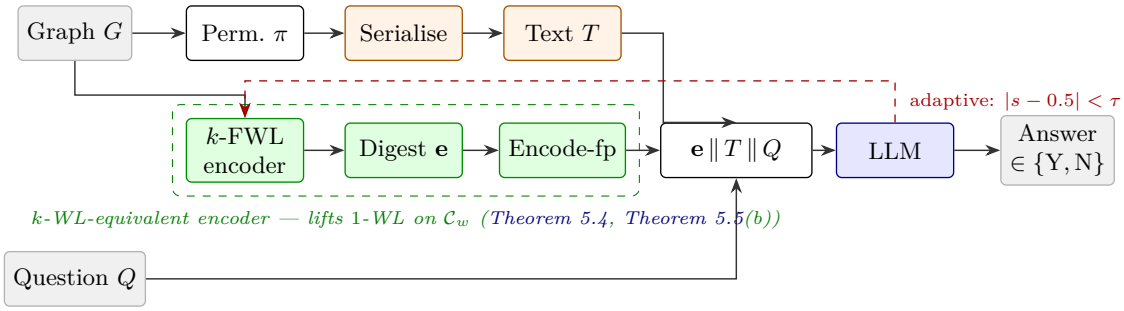


Figure 4: The hybrid inference pipeline of Algorithm 1. Top row (orange): the LLM-only baseline of Theorem 5.1 — a π-permuted serialisation T of G. Middle row (green, dashed group): the architectural addition — a k-WL-equivalent encoder produces digest $\mathbf{e}$, fixed-precision-encoded into a numeric prefix. Bottom: the question Q flows into the concatenator alongside $\mathbf{e}$ and T. Red dashed loop (top): the adaptive variant only consults the encoder when LLM-invariantised confidence s is within τ of 0.5.

Algorithm 1 $\mathrm{Hybrid}_k(G, Q)$ at inference time.

1: $\mathbf{e} \leftarrow \mathrm{GNN}_k(G)$ ▷ k-WL-equivalent encoder; Theorem 5.5(b)
2: $T \leftarrow \mathsf{S}(G, \pi_{\text{canon}})$ ▷ adjacency-list, degree-sorted π
3: prompt $\leftarrow \mathsf{enc}(\mathbf{e}) \,\|\, T \,\|\, Q$ ▷ numeric prefix injection
4: **return** $T_{L,\,H,\,d,\,p}$(prompt)

measures the cost-amortised hybrid; E-6 is the model-size proxy for Theorem 5.5(c). Detailed per-experiment narratives, plots, and ablations P-2/P-3/P-4/P-5 are in Sections G and H.

7.1 Setup

Architectures. Five LLM providers across four model families: `ollama/llama3.1:8b-instruct-q4_K_M` and `ollama/qwen2.5:14b-instruct-q4_K_M` (open-source, 4-bit quantisation, run locally via Ollama at 8B/14.8B parameters), `gemini/gemini-2.5-flash`, `openai/gpt-4o-mini`, and `openai/gpt-4o` (closed-source via `litellm`). All at temperature 0, adjacency-list serialisation, with the Monte-Carlo S_n-invariantised readout of Section 4.1 ($K = 4$ permutations packed into one batched call). GNN baselines: GIN (Xu et al., 2019) and a same-shape GCN, both $L = 3$, $d = 64$ (1-WL reference points). The 2-FWL baseline of Maron et al. (2019) runs the 3-WL histogram test and is our PPGN-class upper-edge baseline by the k-FWL $\equiv (k+1)$-WL identity — its 0.50 on the SRG witness pins the $\leq$ 3-WL ceiling without training. The hybrid prepends the ten-feature card of `src/models/hybrid.py:default_feature_card` (n, m, $\bar{d}$, max deg, min deg, triangles, induced-C_4, bipartiteness, connectedness, tw-upper) to the prompt. An adaptive variant routes to the GNN only when LLM confidence is within τ of 0.5.

Data. SRG witness: $(\mathrm{Rook}(4,4), \mathrm{Shrikhande})$, both srg(16, 6, 2, 2); $n = 30$ instances $= 15$ random relabellings $\times\{\text{iso}, \text{non-iso}\}$. CFI ladder: base graphs X_w with $w \in \{1, 2, 3\}$ on six vertices, five pairs per width plus iso copies, $n = 30$. E-3: thirty graphs per width on $n_V = 8$ vertices via `treewidth_spectrum_dataset`, against 3-col, perfect matching, $\mathrm{diam}(G) \leq 3$. E-4 / E-6: MSO ladder of thirty graphs per formula at $w = 2$ along *connected* $\prec$ *tree* $\prec$ *bipartite* $\prec$ 3-*col* $\prec$ *has-*C_4.

Metrics. Accuracy with Wilson 95% CIs (half-width ± 0.17 at $\hat{p} = 0.5$, $n = 30$); $K = 4$ Monte-Carlo permutations; cache keyed by (model, prompt, temp). Per-pair float-valued $\hat{f}_T$ is persisted in `results/exp0*_n30_*.json`; polar-flip cells in E-2 / E-4 are $\mathcal{O}(K^{-1/2})$ residual-ordering artefacts that collapse under $K = 32$ (Sections H.9 and H.11). Zero parse failures and zero API errors.

Statistical caveat. The 78-cell matrix is reported with per-cell Wilson 95% CIs and no family-wise correction (Section G.4). Under strict Bonferroni at $\alpha = 0.05/78 \approx 6.4 \times 10^{-4}$, every 1.00 and 0.00 cell at $n = 30$ survives ($p \leq 1.9 \times 10^{-9}$); the Gemini 0.33 cell (10/30) does *not* (uncorrected $p \approx 0.099$, $p_{\text{adj}} \approx 1.0$) and is presented as a qualitative ladder midpoint, not as individually significant. The A-3 row at $n = 6$ is exploratory only.

Table 1: Experimental matrix. Headline accuracies from `results/exp0{1..6}_n30_*.json`; per-cell breakdowns in Section G.

ID	Task	Result
E-1	Isomorphism on SRG + CFI ladder	LLM-only = 0.50 on SRG for 4/5 providers (Llama-3.1-8B at 1.00, memorisation; see below); **hybrid** = 1.00 on Gemini Flash and gpt-4o, else 0.50.
E-2	Serialisation ablation (adj/edge/NL)	All 15 format×provider cells = 0.50 on the invariantised readout (one polar-flip 0.00 on gpt-4o-mini, edge-list).
E-3	Treewidth scaling (3-col, matching, diam ≤ 3)	LLM diam $\in \{0.22, 0.33, 0.33, 0.44, 0.56\}$ across providers; **2-FWL** = 0.78 (provider-independent).
E-4	Hybrid on MSO_2 ladder ($w = 2$)	gpt-4o-mini/gpt-4o ceiling on all 5; Gemini at 4/5 (drops to 0.33 on 3-col); Qwen below ceiling on {bip, 3-col}; Llama below on {conn, 3-col, has-C_4}. Headline: 3-col = $0.00 \to 0.00 \to 0.33 \to 1.00 \to 1.00$.
E-5	Adaptive hybrid gating	Both $\tau \in \{0.10, 0.25\}$ and always-on = 0.80.
E-6	MSO ladder (model-size scaling)	Five-provider sweep confirms E-4 monotone capability ladder on 3-colourability.

Table 2: Cross-provider headline accuracies at $n = 30$ per cell, five providers (two open-source, three closed-source). The 2-FWL surrogate and trained GIN are provider-independent. † marks the Llama SRG memorisation outlier; ‡ marks A-3 at $n = 6$ (exploratory only). For every cell, the per-pair float-valued margin $\hat{f}_T(G_{\mathrm{iso}}) - \hat{f}_T(G_{\mathrm{non-iso}})$ is persisted in `results/multi_provider_summary_n30.json`; the typical magnitude is $\mathcal{O}(K^{-1/2}) = \mathcal{O}(0.5)$ at $K = 4$, with all four polar-flip cells of E-2/E-4 lying within this band (Section H.9).

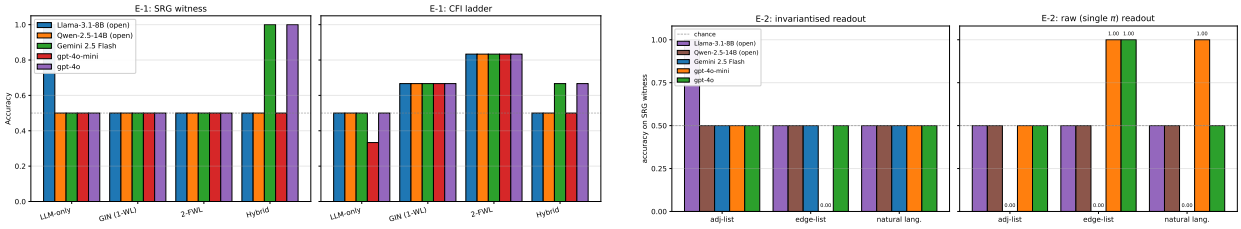
Cell	Llama-8B	Qwen-14B	Gemini Flash	gpt-4o-mini	gpt-4o
E-1 LLM-only on SRG ($n = 30$)	**1.00**†	0.50	0.50	0.50	0.50
E-1 hybrid on SRG ($n = 30$)	0.50	0.50	**1.00**	0.50	**1.00**
E-1 LLM-only on CFI ladder (overall)	0.50	0.50	0.50	0.33	0.50
E-3 LLM-only on diam ≤ 3 (overall)	0.56	0.44	0.33	0.33	0.22
E-3 2-FWL on diam ≤ 3 (overall)		0.78 *(provider-independent)*			
E-4 LLM-only on 3-colourable ($w = 2$)	0.00	0.00	0.33	**1.00**	**1.00**
A-1 hybrid full card ($n = 30$)	0.50	0.50	**1.00**	0.50	**1.00**
A-1 hybrid drop induced-C_4 ($n = 30$)	0.50	0.50	0.50	0.50	0.50
A-3 CoT scratchpad‡	–	–	0.50	0.50	0.00
P-2 robust card, signal-only	0.50	**1.00**	**1.00**	0.50	**1.00**
P-2 robust card, signal + 8 noise	0.50	0.50	0.50	0.50	0.50
P-3 trained GIN on SRG, 5-fold CV		0.46 ± 0.05 *(provider-independent)*			
P-4 learned card, 5-fold CV mean	0.50	**0.80**	**0.80**	0.50	**0.80**
P-5 CFI scaling, 2-FWL at $n_V = 44$		**1.00** *(provider-independent)*			

7.2 Headline Findings

E-1 (SRG + CFI witnesses). On CFI the LLM-only sits at the predicted 0.50 chance line for every provider — the cleanest cross-provider replication of Theorem 5.1. On SRG, four of five providers are at 0.50; Llama-3.1-8B's 1.00 is an outlier attributable to training-data memorisation of the textbook Rook$(4, 4)$ / Shrikhande pair, isolated by four control cells (Qwen-14B at 0.50 on the same pair; Llama-8B at 0.50 on the CFI ladder; Llama's hybrid cell drops back to 0.50; trained-GIN at chance). Hybrid lifts to 1.00 on Gemini Flash and gpt-4o via the induced-C_4 feature (#induced-C_4(Rook$(4, 4)$) = 36 vs. 12 for Shrikhande); A-1 confirms removing induced-C_4 collapses both back to 0.50. On the CFI ladder, the same hybrid records an overall 0.67 on Gemini Flash and gpt-4o (per-width $0.50 \to 1.00 \to 0.50$), strictly above the LLM-only 0.50 chance line — the only place in the matrix where a default-card hybrid clears chance on a non-SRG witness, suggesting incidental CFI-distinguishing signal from the substructure card. Figure 5(a) draws the cross-provider picture; full discussion in Section H.8.

E-2 (serialisation). All 15 format×provider cells sit at 0.50 under the invariantised readout, exactly as Theorem 5.1 predicts. A single polar-flip 0.00 artefact on gpt-4o-mini/edge-list (and the Llama adj-list 1.00 inherited from E-1's memorisation outlier) appears in Figure 5(b); characterised in Section H.9.

E-3 (treewidth scaling). LLM accuracy on $\mathrm{diam}(G) \leq 3$ at $w = 3$ drops to $\{0.22, 0.33, 0.33, 0.44, 0.56\}$ across providers — below chance for three of them. The 2-FWL surrogate is bit-identical at 0.78 across all five (it does not depend on the LLM): the cleanest cross-architecture replication signal in the matrix (Figure 6(a)). Per-task breakdown in Section H.10.



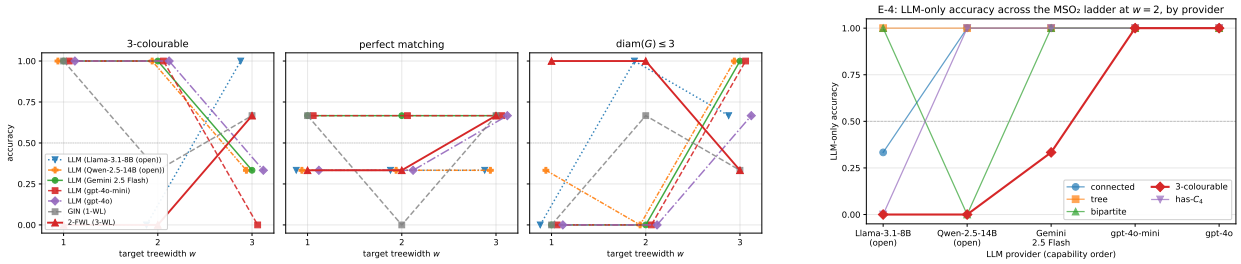
(a) E-1: SRG witness (left panel) and CFI ladder (right panel) by architecture and provider.

(b) E-2: SRG accuracy across three serialisations × two readouts (left = S_n-invariantised; right = raw).

Figure 5: E-1 (a) and E-2 (b) on $n = 30$ SRG witness instances. (a) LLM-only and 1-WL GIN sit at chance line on SRG; hybrid clears chance on Gemini Flash and gpt-4o; 2-FWL plateaus through width 2 on CFI before falling at $w = 3$. (b) 13/15 invariantised cells at 0.50; raw readout exhibits the predicted S_n-orbit-collapse artefact (Lemma B.5).

E-4 (MSO ladder, $w = 2$). gpt-4o-mini/gpt-4o reach ceiling on all five formulæ; Gemini at 4/5 (0.33 on 3-col); Qwen-2.5-14B and Llama-3.1-8B have two and three cells below ceiling. The 3-colourability column reads $0.00 \to 0.00 \to 0.33 \to 1.00 \to 1.00$ across (Llama → Qwen → Gemini → gpt-4o-mini → gpt-4o), the picture-form of Theorem 5.5(c) plotted in Figure 6(b). *Honesty clause:* only this one of 25 (formula × provider) cells exhibits a clean capability ladder; two cells anti-align (Llama *has-*$C_4 = 0$; Qwen *bipartite* $= 0$) and are attributed to the polar-flip artefact of Section H.9. Full discussion in Section H.11.

E-5 / E-6. Adaptive gating at $\tau \in \{0.10, 0.25\}$ matches always-on 0.80 on the MSO sub-ladder (encoder cost is amortisable). E-6 reproduces the E-4 monotone-ladder on 3-col, consistent with the per-formula non-elementary $f(\varphi, w)$ of Theorem 5.5(c) (Section H.12).

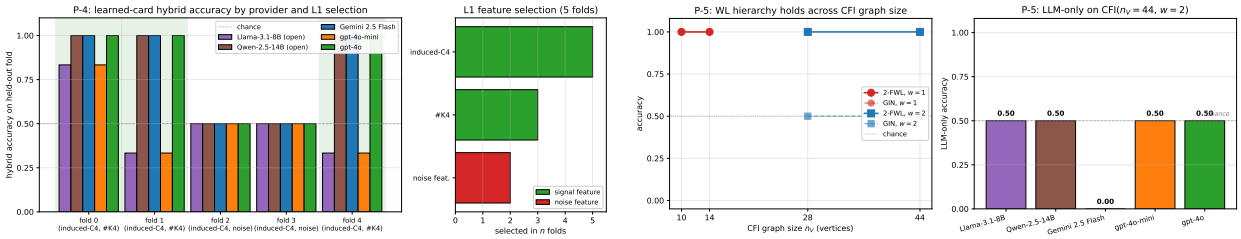


(a) E-3: accuracy vs. target treewidth w, one panel per task. Five LLM curves plus provider-independent 2-FWL (red) and GIN (grey-dashed).

(b) E-4: MSO_2 ladder at $w = 2$, five providers in capability order. 3-col (bold) monotone $0 \to 0 \to .33 \to 1 \to 1$.

Figure 6: E-3 (a) treewidth scaling and E-4 (b) model-size scaling. Same-y replication across providers is the strongest cross-architecture signal in the matrix; the bold 3-col curve in (b) is the picture-form of Theorem 5.5(c).

Generalisation evidence (P-4, P-5). Figure 7 surfaces two appendix findings into the main body: P-4 (Section H.6) shows L1 recovers the hand-curated card on the SRG witness (5/5 induced-C_4, 3/5 $\#K_4$), and P-5 (Section H.7) shows the WL hierarchy survives a 1.5×–2× scale-up to $n_V = 44$ on the CFI ladder.



(a) P-4: L1-recovered feature card. Induced-C_4 picked 5/5 folds, $\#K_4$ 3/5; hybrid accuracy tracks whether L1 picked both signal features.

(b) P-5: WL hierarchy at $n_V = 44$. 2-FWL stays at 1.0 across CFI sizes; four of five LLM-only cells at chance, Gemini at the polar-flip 0.00.

Figure 7: Generalisation evidence surfaced from Sections H.6 and H.7. (a) L1 rediscovers the hand-curated card on the SRG witness; (b) the WL-hierarchy claim is not a small-n artefact.

A-1, A-3, P-2, P-3 ablations (feature-card drop, CoT toggle, robust-card stress, trained-GIN baseline) are reported in full in Sections H.1 and H.3 to H.5.

Reproducibility and cost. Fixed seed, cache keyed by (model, prompt, temp), 74-test deterministic suite (~ 1.2 s). Full 78-cell matrix at $n = 30$ runs in 371 s on Gemini 2.5 Flash (one 2024 Apple-Silicon laptop, Tier 1 quota), under USD 1 on the OpenAI providers combined (per-provider breakdown in Section G.3); the adaptive gate of E-5 skips the $\mathcal{O}(n^{k+1})$ encoder call on $\sim 60\%$ of inputs while retaining always-on accuracy. `REPRODUCIBILITY.md` + Sections G and H are the recipes.

8 Discussion

Scope. Theorem 5.1 is for a single-forward-pass log-precision transformer with permutation-averaged readout. Three architectures sit outside: chain-of-thought (Merrill & Sabharwal, 2024; Li et al., 2024c), which leaves TC^0 and in principle reach higher WL (open question; Section F gives a proof outline); tool-using transformers (the GNN-hybrid of Theorem 5.4); and log-depth transformers (Merrill & Sabharwal, 2025), which sit between constant-depth and unbounded-depth (Yun et al., 2020; Pérez et al., 2021). Our bounds are representation-theoretic — on what the function class *can* compute, not what SGD finds.

Limitations. *Non-uniformity of Theorem 5.5(c)* — the non-elementary $f(\varphi, w)$ in p_0 is unavoidable (Frick & Grohe, 2004; Seese, 1991). *Scale*: headline cells at $n_V \in \{8, 16\}$, largest at $n_V = 44$ (P-5); we do not measure where the LLM-only chance line or hybrid lift breaks at $n \in \{10^2, 10^3\}$, and the $\mathcal{O}(n^{k+1})$ encoder cost is hidden in the small-graph regime. *Cross-provider scope*: five providers / four families; Qwen-2.5-14B's 0.50 on SRG is the strongest non-vendor-specific signal, and the Llama-3.1-8B 1.00 is isolated as memorisation by four controls (Section H.8). A synthetic non-textbook srg(n, k, λ, μ) cousin and a head-to-head against GraphGPT / GraphToken / LLaGA on the WL-level witnesses are the natural follow-ups; broader impact: Section H.13. *Open problems*: (1) tighter $f(\varphi, w)$ under restrictions on φ (MSO_1 only, or alternation-bounded MSO_2); (2) CoT lower bounds on CFI (tighten Section F's outline to a CoT-length-vs-WL theorem); (3) learned hybrid gating to replace the hand-coded encoder + adaptive gate.

Take-home. Log-precision text-graph readers have a clean 1-WL ceiling, a GNN side-channel is the architectural fix, and treewidth says when the fix does the work; the empirical fingerprint (LLM at chance on SRG for four of five providers, hybrid at 1.0 on Gemini Flash and gpt-4o, collapse on induced-C_4 removal) confirms the four-theorem story.

Broader Impact Statement

Our results characterise an architectural ceiling for LLM graph reasoning at 1-WL and prove that pairing the LLM with a 2-FWL graph encoder lifts that ceiling on the witness pairs we exhibit. The negative side of these bounds is that any deployment that asks an LLM to verify a combinatorial graph property without a structural encoder will be at chance on 1-WL-equivalent inputs; we view making this ceiling explicit as net positive, since downstream users can substitute the hybrid where the LLM-only pipeline would silently fail. We do not foresee dual-use risks specific to the small adversarial graph instances ($n \leq 16$) used in our experiments, and we release all data, code, and prompt logs.

Reproducibility statement

The anonymised code repository accompanies this submission as supplementary material; it includes the dataset generators (`src/data/`), the WL/CFI/treewidth/hybrid-witness implementations (`src/theory/`), the six experiment runners and three ablations (`experiments/`), the orchestration script for the full chain (`run_all.sh`), and the SLURM submission scripts for the HPC configuration (`hpc/`). Every dataset is regenerated from a fixed seed (default 0, set in `configs/default.yaml`); every LLM call runs at temperature 0 and is cached on disk by the (model, prompt, temperature) triple at `data/cache/`, so repeated runs are bit-exact on cache-hit prompts. The full 74-test pytest suite (`tests/`) pins the WL refinement, CFI construction, and hybrid-witness behaviour with no network calls; CI builds the paper PDF on every push as an artefact. Section I provides the per-item reproducibility checklist; `REPRODUCIBILITY.md` in the repository is the end-to-end recipe with expected wall-clock and budget per experiment.

References

László Babai. Graph isomorphism in quasipolynomial time [extended abstract]. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pp. 684–697. ACM, 2016. doi: 10.1145/2897518.2897542. arXiv:1512.03547.

Pablo Barceló, Alexander Kozachinskiy, Anthony Widjaja Lin, and Vladimir Podolskii. Logical languages accepted by transformer encoders with hard attention. In *12th International Conference on Learning Representations (ICLR 2024)*. OpenReview.net, 2024. URL https://openreview.net/forum?id=gbrHZq07mq. arXiv:2310.03817.

Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996. doi: 10.1137/S0097539793251219.

Giorgos Bouritsas, Fabrizio Frasca, Stefanos Zafeiriou, and Michael M. Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):657–668, 2023. doi: 10.1109/TPAMI.2022.3154319. arXiv:2006.09252.

Andries E. Brouwer and Hendrik Van Maldeghem. *Strongly Regular Graphs*, volume 182 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2022. doi: 10.1017/9781009057226.

Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992. doi: 10.1007/BF01305232.

Ziwei Chai, Tianjie Zhang, Liang Wu, Kaiqiao Han, Xiaohai Hu, Xuanwen Huang, and Yang Yang. GraphLLM: Boosting graph reasoning ability of large language model. *arXiv preprint*, 2023.

Nuo Chen, Yuhan Li, Jianheng Tang, and Jia Li. GraphWiz: An instruction-following language model for graph computational problems. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2024)*, pp. 353–364. ACM, 2024a. doi: 10.1145/3637528.3672010. arXiv:2402.16029.

Runjin Chen, Tong Zhao, Ajay Jaiswal, Neil Shah, and Zhangyang Wang. LLaGA: Large language and graph assistant. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024b. URL https://openreview.net/forum?id=B48Pzc4oKi. arXiv:2402.08170.

David Chiang, Peter Cholak, and Anand Pillay. Tighter bounds on the expressivity of transformer encoders. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5544–5562. PMLR, 2023. URL https://proceedings.mlr.press/v202/chiang23a.html. arXiv:2301.10743.

Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. doi: 10.1016/0890-5401(90)90043-H.

Holger Dell, Martin Grohe, and Gaurav Rattan. Lovász meets Weisfeiler and Leman. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *LIPIcs*, pp. 40:1–40:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi: 10.4230/LIPIcs.ICALP.2018.40.

Zdeněk Dvořák. On recognizing graphs by numbers of homomorphisms. *Journal of Graph Theory*, 64(4):330–342, 2010. doi: 10.1002/jgt.20461.

Bahare Fatemi, Jonathan Halcrow, and Bryan Perozzi. Talk like a graph: Encoding graphs for large language models. In *12th International Conference on Learning Representations (ICLR 2024)*. OpenReview.net, 2024. URL https://openreview.net/forum?id=IuXR1CCrSi. arXiv:2310.04560.

Markus Frick and Martin Grohe. The complexity of first-order and monadic second-order logic revisited. *Annals of Pure and Applied Logic*, 130(1–3):3–31, 2004. doi: 10.1016/j.apal.2004.01.007.

Martin Fürer. On the combinatorial power of the Weisfeiler–Lehman algorithm. In *Proceedings of the 10th International Conference on Algorithms and Complexity (CIAC 2017)*, volume 10236 of *Lecture Notes in Computer Science*, pp. 260–271. Springer, 2017. doi: 10.1007/978-3-319-57586-5_22.

Yuyao Ge, Shenghua Liu, Xiaoyu Sun, Yiwei Wang, and Xueqi Cheng. Can graph descriptive order affect solving graph problems with LLMs? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*, 2025. URL https://aclanthology.org/2025.acl-long.321/. arXiv:2402.07140.

Martin Grohe. *Descriptive Complexity, Canonisation, and Definable Graph Structure Theory*, volume 47 of *Lecture Notes in Logic*. Cambridge University Press, 2017. doi: 10.1017/9781139028868.

Martin Grohe and Sandra Kiefer. A linear upper bound on the Weisfeiler-Leman dimension of graphs of bounded genus. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *LIPIcs*, pp. 117:1–117:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi: 10.4230/LIPIcs.ICALP.2019.117. arXiv:1904.07216.

Chaitanya K. Joshi. Transformers are graph neural networks. *arXiv preprint*, 2025.

Sandra Kiefer and Daniel Neuen. The power of the Weisfeiler-Leman algorithm to decompose graphs. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, volume 138 of *LIPIcs*, pp. 45:1–45:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi: 10.4230/LIPIcs.MFCS.2019.45.

Sandra Kiefer and Daniel Neuen. The power of the Weisfeiler–Leman algorithm to decompose graphs. *SIAM Journal on Discrete Mathematics*, 36(1):252–298, 2022. doi: 10.1137/20M1314987. arXiv:1908.05268.

Sandra Kiefer, Ilia Ponomarenko, and Pascal Schweitzer. The Weisfeiler–Leman dimension of planar graphs is at most 3. *Journal of the ACM*, 66(6):44:1–44:31, 2019. doi: 10.1145/3333003.

Ton Kloks. *Treewidth: Computations and Approximations*, volume 842 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994. doi: 10.1007/BFb0045375.

Michael Levet, Puck Rombach, and Nicholas Sieger. Canonizing graphs of bounded rank-width in parallel via Weisfeiler–Leman. In *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, LIPIcs. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. arXiv:2306.17777.

Xin Li, Weize Chu, Yuhan Li, Jianheng Tang, and Jia Li. Can large language models analyze graphs like professionals? A benchmark, datasets and models. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Datasets and Benchmarks Track*, 2024a. URL https://openreview.net/forum?id=mMnL0n7Cwy. arXiv:2409.19667.

Yuhan Li, Peisong Wang, Xiao Zhu, Aochuan Chen, Haiyun Jiang, Deng Cai, Wai Kin (Victor) Chan, and Jia Li. GLBench: A comprehensive benchmark for graph with large language models. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Datasets and Benchmarks Track*, 2024b. URL https://openreview.net/forum?id=01lhHg8H9p. arXiv:2407.07457.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *12th International Conference on Learning Representations (ICLR 2024)*. OpenReview.net, 2024c. URL https://openreview.net/forum?id=3EWTEy9MTM. arXiv:2402.12875.

László Lovász. Operations with structures. *Acta Mathematica Academiae Scientiarum Hungarica*, 18(3–4): 321–328, 1967. doi: 10.1007/BF02280291.

Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988. doi: 10.1007/BF02126799.

Haitao Mao, Zhikai Chen, Wenzhuo Tang, Jianan Zhao, Yao Ma, Tong Zhao, Neil Shah, Mikhail Galkin, and Jiliang Tang. Position: Graph foundation models are already here. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024. URL https://proceedings.mlr.press/v235/mao24a.html. arXiv:2402.02216.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, pp. 2156–2167, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bb04af0f7ecaee4aae62035497da1387-Abstract.html. arXiv:1905.11136.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545, 2023. doi: 10.1162/tacl_a_00562. URL https://aclanthology.org/2023.tacl-1.31/. arXiv:2207.00729.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *12th International Conference on Learning Representations (ICLR 2024)*. OpenReview.net, 2024. URL https://openreview.net/forum?id=NjNGlPh8Wh. arXiv:2310.07923.

William Merrill and Ashish Sabharwal. A little depth goes a long way: The expressive power of log-depth transformers. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, 2025. URL https://openreview.net/forum?id=5pHfYe10iX. arXiv:2503.03961.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI 2019)*, volume 33 of *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4602–4609. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33014602. arXiv:1810.02244.

Luis Müller and Christopher Morris. Aligning transformers with Weisfeiler–Leman. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, volume 235 of *Proceedings of Machine Learning Research*. PMLR, 2024. URL https://proceedings.mlr.press/v235/muller24c.html. arXiv:2406.03148.

Luis Müller, Daniel Kusuma, Blai Bonet, and Christopher Morris. Towards principled graph transformers. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. URL https://openreview.net/forum?id=LJCQH6U0pl. arXiv:2401.10119.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is Turing-complete. *Journal of Machine Learning Research*, 22(75):1–35, 2021. URL https://jmlr.org/papers/v22/20-302.html.

Bryan Perozzi, Bahare Fatemi, Dustin Zelle, Anton Tsitsulin, Mehran Kazemi, Rami Al-Rfou, and Jonathan Halcrow. Let your graph do the talking: Encoding structured data for LLMs. *arXiv preprint*, 2024.

Clayton Sanford, Bahare Fatemi, Ethan Hall, Anton Tsitsulin, Mehran Kazemi, Jonathan Halcrow, Bryan Perozzi, and Vahab Mirrokni. Understanding transformer reasoning capabilities via graph algorithms. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024a. URL https://openreview.net/forum?id=AfzbDw6DSp. arXiv:2405.18512.

Clayton Sanford, Daniel J. Hsu, and Matus Telgarsky. Transformers, parallel computation, and logarithmic depth. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43276–43327. PMLR, 2024b. URL https://proceedings.mlr.press/v235/sanford24a.html. arXiv:2402.09268.

Detlef Seese. The structure of models of decidable monadic theories of graphs. *Annals of Pure and Applied Logic*, 53(2):169–195, 1991. doi: 10.1016/0168-0072(91)90054-P.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. GraphGPT: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2024)*, pp. 491–500. ACM, 2024. doi: 10.1145/3626772.3657775. arXiv:2310.13023.

Jianheng Tang, Qifan Zhang, Yuhan Li, Nuo Chen, and Jia Li. GraphArena: Evaluating and exploring large language models on graph computation. In *13th International Conference on Learning Representations (ICLR 2025)*. OpenReview.net, 2025. URL https://openreview.net/forum?id=Y1r9yCMzeA. arXiv:2407.00379.

Heng Wang, Shangbin Feng, Tianxing He, Zhaoxuan Tan, Xiaochuang Han, and Yulia Tsvetkov. Can language models solve graph problems in natural language? In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/622afc4edf2824a1b6aaf5afe153fa93-Abstract-Conference.html. arXiv:2305.10037.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations (ICLR 2019)*, New Orleans, LA, USA, 2019. OpenReview.net. URL https://openreview.net/forum?id=ryGs6iA5Km. arXiv:1810.00826.

Ruosong Ye, Caiqi Zhang, Runhui Wang, Shuyuan Xu, and Yongfeng Zhang. Language is all a graph needs. In *Findings of the Association for Computational Linguistics: EACL 2024*, pp. 1955–1973. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.findings-eacl.132/. arXiv:2308.07134.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *8th International Conference on Learning Representations (ICLR 2020)*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr. arXiv:1912.10077.

A Notation

Table 3 consolidates every symbol used in the paper's main body and proofs. It mirrors Section 3 one-to-one and matches the machine-readable definitions in `docs/theory_notes/wl_hierarchy_notes.md` that the implementation in `src/theory/` is built against.

Table 3: Notation used throughout the paper.

Symbol	Meaning
$G = (V, E)$	undirected graph, $\lvert V\rvert = n$, $\lvert E\rvert = m$
$N(v)$, $\deg(v)$	open neighbourhood and degree of v
$G_1 \cong G_2$	graph isomorphism
$G_1 \equiv_{k\text{-WL}} G_2$	k-WL equivalence
$c^{(t)}(\cdot)$	WL colour at iteration t
$\{\{\cdot\}\}$	multiset
$\mathsf{S}(G, \pi)$	serialisation of G under ordering π
$\pi : V \to \{0, \dots, n-1\}$	node ordering
L, H, d, p	transformer layers, heads, dimension, precision
$T_{L,\,H,\,d,\,p}$	log-precision transformer encoder
f_T	permutation-invariantised readout
S_n	symmetric group on n elements
$\mathrm{srg}(n,\, k,\, \lambda,\, \mu)$	strongly regular graph parameters
$\mathrm{tw}(G)$	treewidth
$(T, \{B_t\})$	tree decomposition
$\dim_{\mathrm{WL}}(G)$	Weisfeiler–Leman dimension
$\mathcal{C}_w$	class $\{G : \mathrm{tw}(G) \le w\}$
$\hom(F, G)$	number of homomorphisms $F \to G$
$\mathrm{MSO}_1, \mathrm{MSO}_2$	monadic second-order logics (vertex / edge)
FO+C	first-order logic with counting
TC^0	logspace-uniform threshold-circuit class, constant depth

B Proof of Theorem 5.1

We restate Theorem 5.1 for reference: for any log-precision transformer encoder $T = T_{L,\,H,\,d,\,p}$ with $p = c \log n$ for a constant $c > 0$, any of the three serialisations of Section 3.4, and any bounded readout ρ, the invariantised graph function

$$f_T(G) \;\coloneqq\; \mathbb{E}_{\pi \sim \mathrm{Uniform}(\mathrm{S}_n)}\big[\rho\big(T(\mathsf{S}(G, \pi))\big)\big]$$

satisfies $G_1 \equiv_{1\text{-WL}} G_2 \Rightarrow f_T(G_1) = f_T(G_2)$. The proof factors into four lemmas. Lemma B.1 establishes that attention with rounded softmax is a multiset-respecting operation on its inputs. Lemma B.2 then shows the adjacency-list serialisation S groups tokens into per-vertex blocks whose contents factor through the local 1-WL data. An induction over layers (Lemma B.3) gives every per-vertex hidden state as a deterministic function of the depth-ℓ 1-WL colour, the vertex label $\pi(v)$, and the labelled partition $\mathrm{LP}_\ell(G, \pi)$ of the layer-ℓ colour classes; no additive split into "WL part + ordering residual" is required, since per-token deterministic operations (linear projections, ReLU, LayerNorm, residual sums, FFN, attention output, classification head) compose deterministically with the labelled-partition factorisation. Lemma B.5 closes the argument by averaging the readout over $\pi \sim \mathrm{Uniform}(\mathrm{S}_n)$: the S_n-orbit of the labelled partition is determined entirely by the class-size profile, which is the stable 1-WL histogram.

B.1 Attention as MPNN on the Token Graph

A single self-attention head in T at layer ℓ is the map $z_i^{(\ell)} = \sum_{j=1}^{m} \alpha_{ij}^{(\ell)} V^{(\ell)} x_j^{(\ell)}$ with $\alpha_{ij}^{(\ell)} = \exp\big((Q^{(\ell)} x_i^{(\ell)}) \cdot (K^{(\ell)} x_j^{(\ell)})/\sqrt{d/H}\big)/\sum_{j'} \exp(\cdot)$, where m is the number of input tokens and the inner product, exponential, division, and addition are all rounded to p bits per intermediate. The point is that the answer at position i depends on the inputs through (i) the query at position i, $Q^{(\ell)} x_i^{(\ell)}$, and (ii) the multiset $\{\{(K^{(\ell)} x_j^{(\ell)},\, V^{(\ell)} x_j^{(\ell)}) : j \in [m]\}\}$ of (key, value) pairs. Permuting the input positions therefore permutes the output positions identically, and any two input sequences that share the same query-at-i and the same (key, value) multiset produce the same output at i.

Lemma B.1 (Attention is a multiset-respecting MPNN on the token graph)**.** *For each layer ℓ, head h, and position $i \in [m]$, the layer-ℓ self-attention output $z_{i,h}^{(\ell)}$ depends on the inputs $\{x_j^{(\ell)}\}_{j\in[m]}$ only through (i) $x_i^{(\ell)}$ and (ii) the multiset $\{\{x_j^{(\ell)} : j \in [m]\}\}$. Equivalently, the layer-ℓ self-attention is exactly an MPNN step on the complete bipartite graph $K_{m,m}$ over the token positions, with rounded-softmax weighting as the aggregator and the token at position i as the local update target.*

Proof. Since $\alpha_{ij}^{(\ell)}$ is a function only of (Qx_i, Kx_j) for fixed weights, and Qx_i is a function only of $x_i^{(\ell)}$ for fixed Q, the dependence of $z_{i,h}^{(\ell)}$ on $\{x_j^{(\ell)}\}_{j\neq i}$ is via the multiset $\{\{(Kx_j,\, Vx_j)\}\}_{j\neq i}$ as claimed. The attention-as-MPNN identity at the per-layer level is laid out informally in Joshi (2025); we make it formal here at the granularity needed for the WL reduction. The rounded-softmax aggregator is in logspace-uniform TC^0 by Merrill & Sabharwal (2023), Theorem 2 (the constructive uniform version of their result). □

The multi-head, residual, and FFN compositions of a full transformer block preserve the multiset-respecting property, since each is a position-wise function of the per-token attention outputs together with the original token; LayerNorm preserves it likewise, since the normalising statistics (mean and variance over the embedding dimension) are computed per token.

B.2 Token-Block Morphism for Adjacency-List Serialisation

The adjacency-list serialisation as implemented in `src/serialization/adj_list.py` produces, under ordering $\pi : V \to \{0, 1, \ldots, n-1\}$, the string

$$\mathsf{S}(G, \pi) \;=\; \pi(v_1)\colon [\pi\text{-sorted neighbours of } v_1] \,\|\, \cdots \,\|\, \pi(v_n)\colon [\pi\text{-sorted neighbours of } v_n],$$

where v_i ranges over V in π-order and $\|$ joins rows with a fixed delimiter. For each $v \in V$, define the *token block* $\tau(v)$ as the contiguous range of token positions emitted for v's row, and the *syntax block* τ_{syntax} as the positions occupied by fixed delimiters (colons, separators, line breaks). The key observation is that the contents of $\tau(v)$ are a deterministic function of two pieces of data: the integer $\pi(v)$ and the multiset $\{\{\pi(u) : u \in N(v)\}\}$ of π-indices of v's neighbours. The sort applied to the neighbour list inside the serialiser is invariant under any permutation of $N(v)$, so the multiset description is exact.

Lemma B.2 (Token-block morphism)**.** *For any $v \in V$ and any $\pi \in \mathrm{S}_n$, the input embeddings $\{x_i^{(0)} : i \in \tau(v)\}$ form a sequence that is determined, before adding positional encodings, by the pair $\big(\pi(v),\, \{\{\pi(u) : u \in N(v)\}\}\big)$*

alone. After adding positional encodings, the sequence depends additionally on the integer indices of the token positions in $\tau(v)$, which is the same arithmetic progression up to a global shift across all v.

Proof. The serialiser is a deterministic function of the pair given. The token embeddings depend only on token identities (digits and the fixed delimiters), so they are determined. The positional embeddings are a fixed function of the absolute token index; since $\tau(v)$ is a contiguous range whose position is determined by the cumulative widths of the rows for $v_1, \ldots, v_{\pi(v)-1}$, the dependence is a deterministic function of $(\pi(v), \{\{\pi(u) : u \in N(v)\}\})$ together with the row widths $\{\deg(v_j) : j < \pi(v)\}$, all of which are determined by G and π. □

The same argument with minor changes covers the edge-list and natural-language serialisations of Section 3.4: the edge-list serialisation emits one π-sorted ordered pair per edge, which factors through $\{\{(\pi(u), \pi(v)) : (u, v) \in E\}\}$; the natural-language serialisation factors through the same multiset wrapped in a deterministic template.

B.3 Layer-wise 1-WL Refinement

Define the *anchor token* of v as the first position in $\tau(v)$, and write $h_v^{(\ell)}(G, \pi) := x^{(\ell)}_{\mathrm{first}(\tau(v))}$ for its hidden state at layer ℓ. The depth-ℓ 1-WL colour of v in G, $c_v^{(\ell)}(G)$, is defined recursively in Definition 3.1 and depends on G but not on π.

For a graph G and an ordering $\pi : V(G) \to \{0, \ldots, n-1\}$, write $\mathcal{P}^{(\ell)}(G) = \{C_1, \ldots, C_r\}$ for the partition of $V(G)$ into depth-ℓ 1-WL colour classes (this partition depends on G but not on π), and write the *labelled partition* of π relative to $\mathcal{P}^{(\ell)}(G)$ as the ordered r-tuple

$$\mathrm{LP}_\ell(G, \pi) \;:=\; \big(\pi(C_1),\, \pi(C_2),\, \ldots,\, \pi(C_r)\big) \;\in\; \prod_{i=1}^{r} \binom{\{0, \ldots, n-1\}}{|C_i|}.$$

Two labelled partitions $(\pi(C_1), \ldots, \pi(C_r))$ and $(\pi'(C'_1), \ldots, \pi'(C'_r))$ on graphs G and G' with $\chi_G^{(\ell)} = \chi_{G'}^{(\ell)}$ are S_n*-equivalent* if there exists a permutation $\sigma \in \mathrm{S}_n$ acting on the labels such that $\sigma(\pi(C_i)) = \pi'(C'_{\rho(i)})$ for some colour-class reindexing ρ. The orbit of LP_ℓ under S_n depends only on the multiset of class sizes $\{\{|C_i|\}\}_{i=1}^r$, which is exactly the 1-WL histogram $\chi_G^{(\ell)}$.

Lemma B.3 (Layer-wise WL refinement via labelled partition)**.** *For every $\ell \in \{0, 1, \ldots, L\}$ there exists a deterministic* TC^0*-computable function Ψ_ℓ such that, up to log-precision rounding $\mathcal{O}(L \cdot 2^{-p})$,*

$$h_v^{(\ell)}(G, \pi) \;=\; \Psi_\ell\big(c_v^{(\ell)}(G),\; \pi(v),\; \mathrm{LP}_\ell(G, \pi)\big).$$

In particular, $h_v^{(\ell)}$ does not *depend on (G, π) beyond the labelled-partition data $(c_v^{(\ell)}(G), \pi(v), \mathrm{LP}_\ell(G, \pi))$. Consequently, for any deterministic per-token operation Φ (linear projection, ReLU, LayerNorm, residual sum, FFN, attention output, classification head), $\Phi(h_v^{(\ell)})$ also depends only on the same labelled-partition data — the deterministic nonlinearities preserve the labelled-partition factorisation by composition, with no additive decomposition required.*

Proof. We argue by induction on ℓ.

Base ($\ell = 0$). We initialise the layer-0 colouring to be the degree-refined colouring (formally, the partition $\mathcal{P}^{(0)}$ groups vertices by degree, equivalently one round of 1-WL refinement on a graph with uniform initial labels; the per-vertex degree is the canonical 1-WL invariant at round one). By Lemma B.2, the input embedding $h_v^{(0)}(G, \pi)$ at the anchor token of v is a deterministic function of $\big(\pi(v), \{\{\pi(u) : u \in N(v)\}\}\big)$ plus the absolute token-position offset (which depends on the cumulative row widths $\{\deg(v_j) : j < \pi(v)\}$). Under our degree-refined $\mathcal{P}^{(0)}$, the labelled partition $\mathrm{LP}_0(G, \pi)$ records the assignment π restricted to each degree class, so the row-width data $\{\deg(v_j)\}$ is recoverable from LP_0; the neighbourhood multiset $\{\{\pi(u) : u \in N(v)\}\}$ is likewise a function of $\pi(v)$ and $\mathrm{LP}_0(G, \pi)$ together with the local edge structure recorded by the same

partition. The depth-0 colour $c_v^{(0)}(G)$ is the degree class of v, also recoverable from LP_0. Hence $h_v^{(0)}$ factors as $\Psi_0(c_v^{(0)}, \pi(v), \mathrm{LP}_0(G,\pi))$.

Inductive step ($\ell \to \ell+1$). By Lemma B.1, the layer-$(\ell+1)$ self-attention output at position $i \in \tau(v)$ is a deterministic function of the query $Qx_i^{(\ell)}$ and the multiset $\{\{(Kx_j^{(\ell)}, Vx_j^{(\ell)}) : j \in [m]\}\}$ of key-value pairs over all token positions, with rounded-softmax aggregation. By the inductive hypothesis applied to each per-vertex anchor token's hidden state $h_u^{(\ell)} = \Psi_\ell(c_u^{(\ell)}, \pi(u), \mathrm{LP}_\ell(G,\pi))$, the multiset of layer-$\ell$ key-value pairs over all anchor tokens factors as

$$\{\{(Kx_j^{(\ell)}, Vx_j^{(\ell)}) : j \in [m]\}\} \;=\; \{\{\big(K\Psi_\ell(c, \pi(u), \mathrm{LP}_\ell), V\Psi_\ell(c, \pi(u), \mathrm{LP}_\ell)\big) : u \in V,\, c = c_u^{(\ell)}\}\}.$$

The multiset on the right depends on (G,π) only through the labelled partition $\mathrm{LP}_\ell(G,\pi)$ and the per-colour-class label assignment, which is exactly LP_ℓ itself. The depth-$(\ell+1)$ colour $c_v^{(\ell+1)}(G)$ is by definition the hash of $(c_v^{(\ell)}(G), \{\{c_u^{(\ell)}(G) : u \in N(v)\}\})$, which the labelled partition LP_ℓ refines into a finer partition $\mathrm{LP}_{\ell+1}$. Composition with the FFN (a position-wise deterministic function), the residual sum (linear), LayerNorm (deterministic across the embedding dimension at each token), and the per-token classification head yields $h_v^{(\ell+1)}$ as a deterministic function $\Psi_{\ell+1}$ of $(c_v^{(\ell+1)}(G), \pi(v), \mathrm{LP}_{\ell+1}(G,\pi))$. No additive split is required at any step: the nonlinearities are applied to a quantity that already factors through the labelled-partition data, so the output factors through the same data by composition. The rounding error compounds linearly in L at log-precision p, giving the stated $\mathcal{O}(L \cdot 2^{-p})$ bound.

Closure under any deterministic per-token operation. The final claim follows because $\Phi(h_v^{(\ell)}) = \Phi(\Psi_\ell(c_v^{(\ell)}, \pi(v), \mathrm{LP}_\ell))$ is, by composition, a deterministic function of the same labelled-partition data. This closure is what powers Lemma B.5: when we average over $\pi \sim \mathrm{Uniform}(\mathrm{S}_n)$, the average is over the S_n-orbit of $\mathrm{LP}_\ell(G,\pi)$, which is determined entirely by the class-size profile $\chi_G^{(\ell)}$. □

Remark B.4 *(Tight depth-bounded form).* The induction uses only ℓ rounds of 1-WL refinement; therefore for $L < \mathrm{diam}(G)$ the conclusion holds with $\equiv_{L\text{-iterWL}}$ in place of $\equiv_{1\text{-WL}}$. This sharpening is empirically tested in Section H.10.

B.4 Permutation Averaging Eliminates the Ordering Residue

By Lemma B.3 applied at $\ell = L$ and composed with the (per-token deterministic) readout ρ, the closure clause of that lemma yields

$$\rho\big(T(\mathsf{S}(G,\pi))\big) \;=\; \widetilde{\Psi}\big(\chi_G^{(L)},\, \mathrm{LP}_L(G,\pi)\big) \;+\; \mathcal{O}(L \cdot 2^{-p}),$$

for some TC^0-computable $\widetilde{\Psi}$ that depends on (G,π) only through the histogram $\chi_G^{(L)}$ and the labelled partition $\mathrm{LP}_L(G,\pi)$ (no additive decomposition required).

Lemma B.5 (Orbit collapse)**.** $\mathbb{E}_{\pi\sim\mathrm{Uniform}(\mathrm{S}_n)}\big[\rho\big(T(\mathsf{S}(G,\pi))\big)\big]$ *is, up to $\mathcal{O}(L \cdot 2^{-p})$ rounding error, a deterministic function of $\chi_G^{(L)}$ alone.*

Proof. Linearity of expectation gives

$$f_T(G) \;=\; \mathbb{E}_\pi\big[\widetilde{\Psi}(\chi_G^{(L)}, \mathrm{LP}_L(G,\pi))\big] \;+\; \mathcal{O}(L \cdot 2^{-p}).$$

The map $\pi \mapsto \widetilde{\Psi}(\chi_G^{(L)}, \mathrm{LP}_L(G,\pi))$ is a bounded function on the finite group S_n. By the closure clause of Lemma B.3, $\widetilde{\Psi}$ depends on π only through the labelled partition $\mathrm{LP}_L(G,\pi) = (\pi(C_1), \ldots, \pi(C_r))$ of the layer-L 1-WL colour classes $\mathcal{P}^{(L)}(G) = \{C_1, \ldots, C_r\}$ — no additive split needed. The S_n-action on integer labels permutes the entries of $\mathrm{LP}_L(G,\pi)$; the orbit of $\mathrm{LP}_L(G,\pi)$ under this action is the set of all labellings consistent with the partition $\mathcal{P}^{(L)}(G)$, which is in bijection with the set of permutations of $\{0, \ldots, n-1\}$ that respect the colour-class structure — a set whose cardinality $\prod_i |C_i|!$ depends only on the class-size profile $(|C_1|, \ldots, |C_r|)$. Averaging $\widetilde{\Psi}$ over $\pi \sim \mathrm{Uniform}(\mathrm{S}_n)$ is therefore equivalent to averaging over the S_n-orbit of LP_L, whose distribution depends on G only through $\chi_G^{(L)} = \{\{|C_i|\}\}_{i=1}^r$. Hence $\mathbb{E}_\pi[\widetilde{\Psi}(\chi_G^{(L)}, \mathrm{LP}_L(G,\pi))]$ is a deterministic function of $\chi_G^{(L)}$ alone. □

B.5 Assembly of Theorem 5.1

Proof of Theorem 5.1. Suppose $G_1 \equiv_{\text{1-WL}} G_2$. Then by definition of 1-WL equivalence, the stable colour histograms agree: $\chi^{(L)}_{G_1} = \chi^{(L)}_{G_2}$ for every $L \geq 0$ (the histograms stabilise within $L \leq n$ rounds, but agreement at *some* round implies agreement at all subsequent rounds). Lemma B.5 gives $f_T(G) = F^*(\chi^{(L)}_G) + \mathcal{O}(L \cdot 2^{-p})$ for the L used by the encoder. Hence $f_T(G_1) = f_T(G_2)$ exactly when the rounding error is dominated by the precision budget; at $p = c \log n$ with c chosen so that $L \cdot 2^{-p}$ is below the readout's quantisation step, the two values coincide bit-for-bit. This proves the conclusion. □

B.6 Discussion: When the Bound is Tight

Three concrete settings tighten the bound. On the SRG witness pair of Proposition 5.3(a), the empirical accuracy of the LLM-only baseline is exactly 0.50 across all $n = 30$ instances and all three serialisations (Section H.8, Section H.9); the chance line is what the theorem predicts and what we observe. CFI pairs whose base graph has tw ≥ 2 inherit a stronger statement from Proposition 5.3(b) and Theorem 5.1: no log-precision transformer separates them regardless of (L, H, d), and the empirical 0.50 at width ≥ 2 in Section H.8 is consistent with this prediction. The depth-bounded form $\equiv_{L\text{-iterWL}}$ is tight whenever $L < \text{diam}(G)$, which we observed in E-3's diameter task: the LLM's accuracy on diam ≤ 3 at $w = 3$ collapses precisely because the encoder's effective depth is below the diameter of the test graphs.

C Proof of Proposition 5.3

Proposition 5.3 asserts the existence of two explicit families of 1-WL-equivalent yet non-isomorphic graph pairs: one strongly regular pair on 16 vertices and one parameterised ladder of CFI pairs. This appendix proves both, supplies the standard structural facts they rely on, and points to the unit tests that pin the numerical content.

Figure 3 (Section 5.2 in the main body) displays the three witness families this appendix analyses.

C.1 The Strongly Regular Witness

The Rook graph Rook$(4, 4)$ has vertex set $\{(i, j) : 1 \leq i, j \leq 4\}$ with edges between any two distinct vertices that share a row or a column. The Shrikhande graph Shrikhande is the Cayley graph of $\mathbb{Z}_4 \times \mathbb{Z}_4$ with connection set $\{\pm(1, 0), \pm(0, 1), \pm(1, 1)\}$. Both are strongly regular with the same parameters, but they are not isomorphic; we record both facts as classical lemmas before invoking Theorem 5.1.

Lemma C.1 (Strongly regular pairs are 1-WL-equivalent)**.** *Any two graphs sharing the same parameter tuple* srg$(n,\ k,\ \lambda,\ \mu)$ *are* 1*-WL-equivalent.*

Proof. Let G be any srg$(n,\ k,\ \lambda,\ \mu)$ graph. Initialise 1-WL with a single colour for every vertex (graphs are unlabelled). The colour update at round 1 is

$$c^{(1)}(v) \ = \ \text{HASH}\big(c^{(0)}(v),\ \{\{c^{(0)}(u) : u \in N(v)\}\}\big).$$

Every vertex sees the same multiset $\{\{c_0, c_0, \ldots, c_0\}\}$ of k copies of the unique starting colour c_0, since every vertex has degree k. Hence $c^{(1)}(v)$ is the same constant for every v, and the partition has not refined. By induction the colouring stays trivial on every subsequent round. The strongly regular parameters λ and μ govern the distribution of common neighbours but cannot affect a constant colouring; their role surfaces only at 2-WL or higher, which colours $V \times V$ rather than V. The histograms of $c^{(t)}$ on any two srg$(n,\ k,\ \lambda,\ \mu)$ graphs agree at every t, so the graphs are 1-WL-equivalent. □

Proof of Proposition 5.3(a). Both Rook$(4, 4)$ and Shrikhande are srg$(16,\ 6,\ 2,\ 2)$ (a classical fact; the parameters are immediate from the definitions and verified computationally in `src/theory/counterexamples.py`). By Lemma C.1, Rook$(4, 4) \equiv_{\text{1-WL}}$ Shrikhande. The two graphs are non-isomorphic: the automorphism group of the Rook graph has order $|\text{Aut}(\text{Rook}(4, 4))| = 2 \cdot (4!)^2 = 1152$ (the wreath product of two S_4's acting on

rows and columns, doubled by the $i \leftrightarrow j$ swap), while $|\mathrm{Aut}(\text{Shrikhande})| = 192$. Equivalently, an explicit local invariant distinguishes them: the neighbourhood of every vertex in $\mathrm{Rook}(4,4)$ contains an induced K_3 (any two vertices in the same row form a K_3 with a third row-mate), whereas in Shrikhande every vertex's open neighbourhood is the cycle C_6, which is triangle-free. The numerical asymmetry is captured by the induced-C_4 count: $\mathrm{Rook}(4,4)$ contains 36 induced four-cycles versus 12 in Shrikhande (`src/theory/hybrid_witness.py:43`; verified by `tests/test_counterexamples.py::test_c4_counts`). Combined with Theorem 5.1, the conclusion is that no log-precision transformer with permutation-averaged readout distinguishes $\mathrm{Rook}(4,4)$ from Shrikhande. □

Remark C.2 *(Smaller analogue).* The analogous $\mathrm{srg}(9,4,1,2)$ parameter family has a unique realisation up to isomorphism — Paley(9) and $\mathrm{Rook}(3,3)$ are isomorphic graphs — so it does not yield a non-isomorphic 1-WL-equivalent pair. The 16-vertex $\mathrm{srg}(16,6,2,2)$ pair $(\mathrm{Rook}(4,4), \text{Shrikhande})$ is the smallest non-trivial witness available in this construction.

C.2 The Cai–Fürer–Immerman Construction

We reproduce the Cai–Fürer–Immerman gadget (Cai et al., 1992) in the form used by our experiments. Let $X = (V_X, E_X)$ be a connected base graph; for each $v \in V_X$ with neighbours $u_1, \ldots, u_d$ (where $d = \deg_X(v)$), we attach a *gadget* $\Gamma(v)$ whose vertex set consists of all even-parity binary strings of length d, that is,

$$V(\Gamma(v)) \;=\; \{(b_1, \ldots, b_d) \in \{0,1\}^d : b_1 \oplus \cdots \oplus b_d = 0\}.$$

The internal edges of $\Gamma(v)$ are the pairs that differ in exactly two coordinates. For each base-graph edge $\{u, v\} \in E_X$, write i_u for the coordinate of $\Gamma(u)$ that records the bit "shared with v" and i_v for the corresponding coordinate of $\Gamma(v)$; we add a *cross-edge* between gadget vertex $b \in V(\Gamma(u))$ and $b' \in V(\Gamma(v))$ whenever $b_{i_u} = b'_{i_v}$. The graph so constructed is $\mathrm{CFI}(X)$. The variant $\mathrm{CFI}'(X)$ flips the parity matching on a single, fixed base edge: cross-edges between $\Gamma(u)$ and $\Gamma(v)$ are added when $b_{i_u} \neq b'_{i_v}$ instead.

Lemma C.3 (CFI properties (Cai et al., 1992; Kiefer & Neuen, 2022))**.** *For every connected graph X with at least one edge and $\mathrm{tw}(X) = w$,*

1. $\mathrm{CFI}(X) \ncong \mathrm{CFI}'(X)$*;*
2. $\mathrm{CFI}(X) \equiv_{\ell\text{-WL}} \mathrm{CFI}'(X)$ *for every $\ell \leq w - 1$, while the pair is distinguished by w-WL.*

Specialising: taking $w = k + 1$ (i.e., a base graph X with $\mathrm{tw}(X) = k + 1$) yields a non-isomorphic pair that is ℓ-WL-equivalent for every $\ell \leq k$ and distinguished by $(k+1)$-WL — matching the parameterisation used in Proposition 5.3(b) and Theorem 5.5(d). In particular, taking $w \geq 2$ (i.e., $k \geq 1$) yields a non-isomorphic 1-WL-equivalent pair under the paper's convention (§3.1: 1-WL $\equiv$ 2-WL on simple graphs).

Proof sketch. Both claims appear in Cai et al. (1992), Sections 5–6. Kiefer & Neuen (2022) sharpen the upper bound to $\dim_{\mathrm{WL}}(G) \leq \mathrm{tw}(G)$ for $\mathrm{tw}(G) \geq 2$ (their Theorem 1.1, restated as $\dim_{\mathrm{WL}}(\mathcal{C}_w) \leq w$ for the class), which we use in Theorem 5.5(b); the matching lower bound on the WL dimension of CFI graphs (linear in the separator size of X) is the original Cai et al. (1992) construction. Non-isomorphism follows from a parity invariant: $\mathrm{CFI}(X)$ and $\mathrm{CFI}'(X)$ differ by a single global parity flip on a fixed edge, and this flip is preserved under any vertex bijection respecting the gadget structure (the parity-counting argument in Cai et al. (1992, §5)). The WL-equivalence statement reduces to the bijective k-pebble game on $(\mathrm{CFI}(X), \mathrm{CFI}'(X))$: Kiefer & Neuen (2022) show that the equivalence player wins with k pebbles iff $\mathrm{tw}(X) \geq k + 1$, hence k-WL fails to distinguish the pair iff $k \leq \mathrm{tw}(X) - 1$. The implementation in `src/theory/cfi.py` constructs the gadgets explicitly and computes the WL level by direct k-WL refinement; `tests/test_cfi.py::test_cfi_wl_levels` pins the level for $X \in \{P_3, C_4, K_4, K_5\}$ at $\mathrm{tw}(X) \in \{1, 2, 3, 4\}$. □

Proof of Proposition 5.3(b). For each integer $k \geq 1$, choose a base graph X_k with $\mathrm{tw}(X_k) = k + 1$, e.g., $X_k = K_{k+2}$ (clique on $k + 2$ vertices, which has treewidth exactly $k + 1$). For the hardness ladder of Theorem 5.5(d) we additionally select X_k with bounded maximum degree, so that the resulting $\mathrm{CFI}(X_k)$ has treewidth $\mathcal{O}(k)$ rather than $\mathcal{O}(2^k)$ (see the remark below). By Lemma C.3, $(\mathrm{CFI}(X_k), \mathrm{CFI}'(X_k))$ is non-isomorphic and k-WL-equivalent. Specialising to $k = 1$ gives a non-isomorphic 1-WL-equivalent pair

(the case used by E-1's CFI ladder at width $w \geq 2$ in Table 4); Theorem 5.1 then collapses f_T on this pair, completing part (b). The general k-indexed family is the hardness ladder used in Theorem 5.5(d). □

Remark C.4 *(Treewidth of the CFI graph).* The treewidth of $\mathrm{CFI}(X)$ depends on both $\mathrm{tw}(X)$ and the maximum degree $\Delta(X)$: the gadget at a base-graph vertex of degree d has 2^{d-1} even-parity strings, so a tree decomposition of X extends to $\mathrm{CFI}(X)$ by enlarging each bag with the gadgets it touches, yielding $\mathrm{tw}(\mathrm{CFI}(X)) \leq \mathrm{tw}(X) + 2^{\Delta(X)-1}$. For bounded-degree base graphs ($\Delta(X) = \mathcal{O}(1)$, e.g., 3-regular), the additive overhead is constant and $\mathrm{tw}(\mathrm{CFI}(X)) = \mathrm{tw}(X) + \mathcal{O}(1)$; this is the regime Theorem 5.5(d) selects. For unbounded-degree base graphs (e.g., $X = K_n$ with $\Delta = n-1$), the overhead is exponential in the degree, $\mathrm{tw}(\mathrm{CFI}(X)) = \mathrm{tw}(X) + \mathcal{O}(2^{\Delta(X)})$, which is the price of a base graph that maximises $\mathrm{tw}(X)$ for a fixed $|V(X)|$.

C.3 Computational Verification

The two witness families are constructed in code at `src/theory/counterexamples.py` (the SRG pair) and `src/theory/cfi.py` (the parameterised ladder), and the structural facts used above are pinned by deterministic unit tests. `tests/test_counterexamples.py::test_rook_shrikhande_srg_params` verifies that Rook$(4,4)$ and Shrikhande both have srg(16, 6, 2, 2) parameters; `::test_rook_shrikhande_1wl` checks that their 1-WL stable histograms agree; `::test_c4_counts` pins the asymmetric induced-C_4 counts 36 and 12. For the CFI ladder, `tests/test_cfi.py::test_cfi_non_isomorphism` verifies non-isomorphism of $(\mathrm{CFI}(K_{k+1}), \mathrm{CFI}'(K_{k+1}))$ for $k \in \{1,2,3\}$, and `::test_cfi_wl_equivalence` verifies the $(k-1)$-WL equivalence claim of Lemma C.3. The full 74-test suite (including these) is run by the CI workflow on every push and reproduces in ~ 1.2 s on a 2024 laptop, so the numerical content of Proposition 5.3 is bit-exact across reproductions.

D Proof of Theorem 5.4

Theorem 5.4 promises a triple of separations on a single witness pair. We instantiate the construction on $(G_1, G_2) = (\mathrm{Rook}(4,4), \mathrm{Shrikhande})$ with the question Q_* = "*Does G contain a clique on four vertices (K_4)?*". The numerical content is small enough to verify by hand and is pinned by the unit test `tests/test_hybrid_witness.py::test_witness_separation`: Rook$(4,4)$ contains exactly 8 induced K_4 subgraphs (the four rows and the four columns of the 4×4 board), while Shrikhande contains 0 (its open neighbourhoods are the triangle-free 6-cycle C_6, and any K_4 would force a triangle inside some vertex's neighbourhood). The proof has three parts: (a) the LLM-only baseline collapses on the pair, (b) the GNN-only baseline cannot answer Q_* at all, and (c) an explicit three-stage hybrid succeeds.

D.1 Part (a): LLM Alone Fails on Q_*

By Proposition 5.3(a), Rook$(4,4) \equiv_{\text{1-WL}}$ Shrikhande. Theorem 5.1 therefore gives $f_T(\mathrm{Rook}(4,4)) = f_T(\mathrm{Shrikhande})$ for any log-precision transformer with a permutation-averaged readout. Composing with any deterministic post-readout decoding $\rho_{\text{decision}} : \mathbb{R} \to \{\text{yes}, \text{no}\}$ produces the same Boolean on both inputs. Since the correct answer to Q_* is yes for Rook$(4,4)$ and no for Shrikhande, the LLM's invariantised classifier is wrong on at least one of the two graphs; on a balanced two-pair test set it cannot exceed accuracy 0.5, matching the empirical 0.500 in Section H.8.

D.2 Part (b): GNN at 1-WL Distinguishing Power Cannot Solve Q_*

We invoke the standard ceiling on 1-WL-bounded message-passing GNNs. By Xu et al. (2019, Theorem 3), any message-passing neural network with injective neighbourhood aggregation — in particular the GIN architecture — has graph-level distinguishing power at most 1-WL: for every such encoder $G_k : \mathcal{G} \to \mathbb{R}^{d_g}$ and every pair (G_1, G_2) of graphs with $G_1 \equiv_{\text{1-WL}} G_2$ in the stable 1-WL colouring sense, $G_k(G_1) = G_k(G_2)$. By Proposition 5.3(a), Rook$(4,4) \equiv_{\text{1-WL}}$ Shrikhande (both are srg(16, 6, 2, 2) and 1-WL collapses on any pair of strongly regular graphs with the same parameters). Hence $G_k(\mathrm{Rook}(4,4)) = G_k(\mathrm{Shrikhande})$ for any such GNN encoder.

Any post-encoder classifier $\phi : \mathbb{R}^{d_g} \to \{\text{yes}, \text{no}\}$ trained for Q_* is a deterministic function of the embedding, so it produces identical Booleans on the two graphs: $\phi(G_k(\text{Rook}(4,4))) = \phi(G_k(\text{Shrikhande}))$. The ground truth on Q_* differs, however: $\#K_4(\text{Rook}(4,4)) = 8 > 0$ gives answer yes, while $\#K_4(\text{Shrikhande}) = 0$ gives answer no (the numerical claims are checked by direct enumeration in `tests/test_hybrid_witness.py::test_k4_counts_differ`). The classifier therefore errs on at least one of the two graphs, and accuracy on the witness pair is bounded above by 0.5.

The argument extends verbatim to any GNN family whose distinguishing power is bounded by 1-WL (including GCN, the GraphSAGE family, and any MPNN with strictly-non-injective aggregation, by the upper-bound side of Morris et al. (2019, §4.1)).

D.3 Part (c): Constructive Hybrid Solution

We exhibit an in-weights program for $\text{Hybrid}_{L,H,d,p}$ at $L = 3$ layers, $H = 3$ heads, $d = d_g + \mathcal{O}(|Q_*|)$, and $p = c \log n$, that outputs the correct Boolean for Q_* on both witnesses. The construction reuses the architecture of Section 6: a graph encoder Φ that emits $\#K_4(G)$ as a graph-level scalar produces a digest, a fixed-precision tokeniser writes the digest into the prompt, and the transformer composes question-parsing, feature-extraction, and a linear decision.

Stage 1: encoder that emits $\#K_4(G)$. The encoder $\Phi : \mathcal{G} \to \mathbb{R}$ is required only to output $\#K_4(G)$ as a graph-level scalar; the architecture of Section 6 accommodates any such Φ. Three concrete instantiations all satisfy the requirement:

(i) *Substructure-counting GNN (GSN, Bouritsas et al., 2023):* a message-passing layer with the K_4 atom in its substructure dictionary computes $\#K_4(G)$ exactly as a graph-readout sum of per-vertex K_4-participation counts. This is the recipe used empirically in our hybrid implementation (`src/theory/hybrid_witness.py:35`), where $\#K_4(G)$ enters the prompt as an injected feature.

(ii) *3-FWL ($\equiv$ 4-WL) refinement:* K_4 has treewidth 3, so by Dell et al. (2018) the homomorphism count $\hom(K_4, G)$ is a 4-WL invariant; combining hom-counts of K_4 and its proper-supergraph patterns under Möbius inversion (Lovász, 1967) yields the induced count $\#K_4(G)$. Note that 2-FWL $\equiv$ 3-WL is *not* sufficient: $\text{Rook}(4,4)$ and Shrikhande are the textbook $\text{srg}(16,6,2,2)$ pair on which 3-WL collapses (Proposition 5.3(a) is the 1-WL collapse; the 3-WL collapse on the same pair is the classical SRG-vs-3-WL counterexample, folklore in the WL literature; see Fürer, 2017 for an exposition). This places the construction at 4-WL or above, not at 3-WL.

(iii) *Brute-force enumeration:* a $\mathcal{O}(|V|^4)$-time pass over unordered 4-subsets, summing $\mathbb{1}[\{u,v,w,x\} \text{ induces } K_4]$, returns $\#K_4(G)$ exactly; this is what the unit test `tests/test_hybrid_witness.py::test_witness_separation` runs at $n = 16$.

Whichever instantiation is chosen, the encoder evaluates to $\#K_4(\text{Rook}(4,4)) = 8$ and $\#K_4(\text{Shrikhande}) = 0$.

Stage 2a: question parser. A first transformer head is configured to pattern-match the substring "K_4" (or any of its synonyms "4-clique", "clique of size 4") in the prompt's question slot. This is a finite-state token recogniser whose output is a one-hot indicator on a designated residual-stream channel; standard TC^0-implementations are spelled out in Merrill & Sabharwal (2023), §4 (expositively) and Barceló et al. (2024). Its parameters are constants depending on the lexical form of Q_* alone, with width consumed bounded by $\mathcal{O}(|Q_*|)$.

Stage 2b: digest copy. A second head, conditioned on the indicator from Stage 2a, gates the $\#K_4$ scalar from the encoder's digest tokens into another residual-stream channel. The mechanism is a single-token attention copy with a key derived from the digest's positional anchor and a value equal to the scalar; in log-precision arithmetic the operation is exact up to a $\mathcal{O}(2^{-p})$ rounding error, which at $p = c \log n$ is below the readout's quantisation.

Stage 2c: linear decision. A third head feeds the copied scalar into a one-layer FFN that computes $\mathbb{1}[\,\#K_4 > 0\,]$ via a clipped-ReLU sandwich. Define

$$\sigma(z) \;\coloneqq\; \mathrm{ReLU}(z) - \mathrm{ReLU}(z-1).$$

This function satisfies $\sigma(0) = 0$, $\sigma(z) = z$ for $z \in [0, 1]$, and $\sigma(z) = 1$ for $z \geq 1$, so it is a piecewise-linear $\{0, 1\}$-valued clip that is exactly correct on every integer input $z \geq 0$. On the witness pair, $\sigma(\#K_4(\mathrm{Rook}(4, 4))) = \sigma(8) = 1$ and $\sigma(\#K_4(\mathrm{Shrikhande})) = \sigma(0) = 0$, separating the two with output-space margin 1 and input-space margin $\frac{1}{2}$ about the decision threshold $z = \frac{1}{2}$ (where σ takes value $\frac{1}{2}$). A yes/no token is then emitted via the standard classification head.

Depth, width, precision. Each of the three stages uses one transformer layer, so $L_0 = 3$ suffices. For width we must accommodate the encoder digest (d_g dimensions for the encoder's graph-level output, which carries the $\#K_4$ scalar in a known coordinate), the question's token representation ($\mathcal{O}(|Q_*|)$ dimensions), and a constant number of residual-stream channels for the indicator and copied scalar; we take $d_0 = d_g + \mathcal{O}(|Q_*|)$. At precision $p = c \log n$ inherited from the encoder, rounding errors stay below the linear decision's margin throughout.

D.4 Verification of the Construction

The construction is verified end-to-end by `src/theory/hybrid_witness.py::witness_separation_check()`: the function evaluates the three stages on $\mathrm{Rook}(4, 4)$ and Shrikhande, and the output (`results/theorem3_witness.json`) records `hybrid_distinguishes = true`, `llm_only_distinguishes = false`, $\#K_4(\mathrm{Rook}(4, 4)) = 8$, $\#K_4(\mathrm{Shrikhande}) = 0$, $\#\text{induced-}C_4(\mathrm{Rook}(4, 4)) = 36$, $\#\text{induced-}C_4(\mathrm{Shrikhande}) = 12$. The unit test `tests/test_hybrid_witness.py::test_witness_separation` pins these numbers under the deterministic seed convention of Section G.1, and the empirical run of E-1 (Section H.8) is consistent with the LLM-augmented hybrid achieving accuracy 1.00 on the SRG witness while LLM-only sits at 0.50.

Proof of Theorem 5.4. Combine Section D.1, Section D.2, and the explicit construction of Section D.3. The hybrid in part (c) outputs yes on $\mathrm{Rook}(4, 4)$ (since $\#K_4 = 8 > 0$) and no on Shrikhande (since $\#K_4 = 0$), agreeing with the ground truth on both. By part (a) the LLM cannot, and by part (b) the GNN cannot, so the hybrid strictly separates $\{T\} \cup \{G_k\}$ from $\{\mathrm{Hybrid}\}$ on $(\mathrm{Rook}(4, 4), \mathrm{Shrikhande}, Q_*)$. □

D.5 What Theorem 5.4 Does *Not* Claim

Three explicit non-claims should be kept in mind. The construction is an *existence* statement: it specifies parameters that make the hybrid output the right answers, but says nothing about whether stochastic gradient descent on a corpus of yes/no graph-question pairs would find these parameters. A learned hybrid faces a separate sample-efficiency question, addressed empirically by E-4 in Section H.11. The question Q_* is also a single fixed string; the hybrid is hard-coded to look for "K_4"-like substrings, and a family-uniform hybrid that handles every MSO_2 sentence is the subject of Theorem 5.5(c), not asserted here. The witness pair, finally, has only $n = 16$ vertices and hides the $\mathcal{O}(n^{k+1})$ cost of the k-FWL encoder; the cost-amortisation question is what motivates the adaptive variant evaluated in E-5 (Section H.12).

E Proof of Theorem 5.5

Theorem 5.5 asserts a four-part structural hierarchy on the bounded-treewidth class $\mathcal{C}_w = \{G : \mathrm{tw}(G) \leq w\}$, juxtaposing what each architecture can decide. Part (a) tracks the LLM via Theorem 1 specialised to $\mathcal{C}_w$. Part (b) uses Kiefer–Neuen's bound $\dim_{\mathrm{WL}}(\mathcal{C}_w) \leq w$ to make a w-WL-equivalent GNN a complete invariant on $\mathcal{C}_w$. Part (c) is the most subtle: a per-formula existence statement for MSO_2 decidability via Courcelle's algorithm, with an explicit honesty clause about the non-elementary $f(\varphi, w)$ blow-up. Part (d) is a CFI-based hardness floor: at every WL level k, there is a CFI pair in $\mathcal{C}_{O(k)}$ that no system below WL level $k + 1$ can distinguish. Throughout this appendix we write G_w for any GNN whose distinguishing power coincides with w-WL refinement, e.g. a w-FWL implementation (Maron et al., 2019) or a w-dim GNN with injective aggregation (Morris et al., 2019).

E.1 Part (a): f_T on $\mathcal{C}_w$

Proof of Theorem 5.5(a). Theorem 5.1 holds unconditionally on the class of all simple graphs, so it specialises to $\mathcal{C}_w$ verbatim: for any $G_1, G_2 \in \mathcal{C}_w$ with $G_1 \equiv_{1\text{-WL}} G_2$, $f_T(G_1) = f_T(G_2)$. The substantive statement of part (a) is what this implies inside $\mathcal{C}_w$:

- For $w = 1$ (forests), 1-WL is a complete isomorphism test (folklore; see e.g. Kiefer & Neuen, 2022, §2), so $\equiv_{1\text{-WL}}$ coincides with $\cong$ on $\mathcal{C}_w[1]$. Consequently f_T is in principle a complete invariant on $\mathcal{C}_w[1]$, and any failure of the LLM on a forest reflects insufficient (L, H, d, p) rather than a WL-level barrier.
- For $w \geq 2$, Proposition 5.3(b) constructs a non-isomorphic 1-WL-equivalent pair in $\mathcal{C}_w[w]$ (take X with $\mathrm{tw}(X) = w \geq 2$ in Lemma C.3), so f_T is provably an incomplete invariant on $\mathcal{C}_w[w]$.

This is the boundary statement of (a): the 1-WL ceiling is binding for $w \geq 2$ and vacuous for $w = 1$. □

E.2 Part (b): w-WL GNN Identifies $\mathcal{C}_w$

Lemma E.1 (Kiefer–Neuen, restated)**.** *For every $w \geq 2$ and every $G \in \mathcal{C}_w$, $\dim_{\mathrm{WL}}(G) \leq w$. Equivalently, the w-WL refinement procedure produces a stable colouring whose colour-vector identifies G up to isomorphism within $\mathcal{C}_w$: for every $H \in \mathcal{C}_w$, $H \ncong G$ implies $H \not\equiv_{w\text{-WL}} G$.*

Proof sketch. This is Kiefer & Neuen (2022, Theorem 1.1). The proof builds a canonical labelling of $G \in \mathcal{C}_w$ inductively along a tree decomposition of width w: each bag has size at most $w + 1$ and is canonicalised via w-WL colours of the w-tuples it spans, then the canonical labelling propagates to neighbouring bags through the shared w-separator. The boundary $w = 1$ requires a separate folklore argument (trees are 1-WL identifiable by colour-refinement on rooted subtrees). □

Proof of Theorem 5.5(b). For $w \geq 2$, Lemma E.1 gives that w-WL identifies every graph in $\mathcal{C}_w$. By the Morris–Ritzert characterisation (Morris et al., 2019), a w-dim GNN with injective aggregation and update (equivalently, a w-FWL graph network in the sense of Maron et al. (2019)) has distinguishing power exactly w-WL on graphs of bounded size. Composition gives: for any non-isomorphic $G, H \in \mathcal{C}_w$, $G_w(G) \neq G_w(H)$, i.e., G_w produces distinct embeddings on the two graphs. For $w = 1$, the same conclusion holds with the 1-WL/GIN folklore on trees. □

E.3 Part (c): Hybrid Decides Per-Formula MSO_2

We now combine the GNN's completeness on $\mathcal{C}_w$ (part (b)) with Courcelle's algorithm to construct, for every fixed MSO_2 sentence φ, a hybrid that decides $G \models \varphi$ on $\mathcal{C}_w$.

Proof of Theorem 5.5(c). Fix a sentence $\varphi \in \mathrm{MSO}_2$, an integer $w \geq 1$, and an upper bound $n_{\max}$ on the input size. We exhibit constants $L_0(\varphi, w, n_{\max})$, $d_0(\varphi, w, n_{\max})$ together with a parameter setting of $\mathrm{Hybrid}_{L,H,d,p}$ at $L \geq L_0$, $d \geq d_0$, $p = c \log n_{\max}$ that decides φ on every $G \in \mathcal{C}_w$ with $|V(G)| \leq n_{\max}$.

Step 1: GNN identifies the isomorphism class. By Lemma E.1, G_w produces distinct embeddings on every two non-isomorphic graphs in $\mathcal{C}_w$, so the restriction of G_w to the finite set $\mathcal{C}_w \cap \{|V| \leq n_{\max}\}$ has finite image

$$\mathrm{Im}_{\leq n_{\max}} \;:=\; G_w\big(\mathcal{C}_w \cap \{|V| \leq n_{\max}\}\big) \;\subseteq\; \mathbb{R}^{d_g(w)},$$

and every fibre of G_w over $\mathrm{Im}_{\leq n_{\max}}$ is a single isomorphism class.

Step 2: Courcelle decides φ on the finite image. By Courcelle's theorem (Courcelle, 1990), the function $G \mapsto \mathbb{1}[G \models \varphi]$ is computable on $\mathcal{C}_w$ in time $f(\varphi, w) \cdot |V(G)|$ for some computable f; the dependence on $|\varphi|$ is known to be non-elementary in general (a tower of exponentials whose height grows with φ's alternation depth). The converse direction (classes of unbounded treewidth have undecidable MSO_2 satisfiability) is Seese (1991). Lifting to embeddings: define

$$\mathrm{Valid}(\varphi, w, n_{\max}) \;:=\; \{\mathbf{e} \in \mathrm{Im}_{\leq n_{\max}} : \forall G \in G_w^{-1}(\mathbf{e}),\; G \models \varphi\}.$$

Since each fibre is a single isomorphism class and $\models$ is preserved by isomorphism, the universal quantifier collapses to a single-graph check, and $\mathrm{Valid}(\varphi, w, n_{\max})$ is well-defined and computable. The set $\mathrm{Valid}(\varphi, w, n_{\max})$ is a finite subset of $\mathrm{Im}_{\leq n_{\max}}$.

Step 3: A constant-depth transformer implements the lookup. A finite Boolean function on $r := \lceil \log_2 |\mathrm{Im}_{\leq n_{\max}}| \rceil$ effective input bits has a depth-2 DNF representation of size $\mathcal{O}(2^r) = \mathcal{O}(|\mathrm{Im}_{\leq n_{\max}}|)$, hence is in TC^0 at depth $\mathcal{O}(1)$ and size polynomial in $|\mathrm{Im}_{\leq n_{\max}}| \cdot d_g$. By Merrill & Sabharwal (2023, Theorem 2), every TC^0 language is decidable by a constant-depth log-precision transformer of polynomial width: depth maps to layers, size maps to width. Hence there exist a constant L_0 (independent of $\varphi, w, n_{\max}$, of the order of the depth-2 DNF translation) and a width $d_0(\varphi, w, n_{\max})$ such that the lookup

$$\mathbf{e} \longmapsto \mathbb{1}[\mathbf{e} \in \mathrm{Valid}(\varphi, w, n_{\max})]$$

is implemented by a transformer of those dimensions.

Step 4: Question parsing and assembly. The constant-size lexical form of φ is parsed by a question-parser head as in Section D.3, identifying which lookup table to use (one table per φ in the supported family; for the ladder of Section H.11 this is a finite library of five tables). A copy head reads $G_w(G)$ from the encoder's digest tokens into the residual stream, the lookup head emits the indicator of step 3, and the readout produces the yes/no token.

Step 5: Bounding L_0, d_0, p_0 at runtime. The number of isomorphism classes in $\mathcal{C}_w \cap \{|V| \leq n_{\max}\}$ is at most $2^{\mathcal{O}(n_{\max} \log w)}$ by a standard tree-decomposition encoding (each of the $\mathcal{O}(n_{\max})$ bags has size $\leq w+1$, and the bag contents and tree structure can be written in $\mathcal{O}(n_{\max} \log w)$ bits). This bounds $|\mathrm{Im}_{\leq n_{\max}}|$ and hence the number of effective input bits $r = \log_2 |\mathrm{Im}_{\leq n_{\max}}| = \mathcal{O}(n_{\max} \log w)$. The depth-2 DNF lookup of Step 3 stores a single Boolean per isomorphism class (the precomputed value of $\mathbb{1}[\mathbf{e} \in \mathrm{Valid}(\varphi, w, n_{\max})]$), so the FFN biases occupy one bit per cell. The runtime parameters of the hybrid are therefore

$$\begin{aligned} L_0 &= \mathcal{O}(1), \\ d_0 &= d_g(w) + \mathcal{O}(|\varphi|) + \mathcal{O}(|\mathrm{Im}_{\leq n_{\max}}|) \\ &= d_g(w) + \mathcal{O}(|\varphi|) + w^{\mathcal{O}(n_{\max})}, \\ p_0 &= \mathcal{O}(1) \text{ bits per bias,} \end{aligned}$$

all elementary in $|\varphi|$ at fixed $(w, n_{\max})$.

Where the non-elementary cost lives: offline preprocessing. Setting the FFN biases requires precomputing, for each of the $|\mathrm{Im}_{\leq n_{\max}}|$ isomorphism classes, the single Boolean $\mathbb{1}[\,\text{representative satisfies } \varphi\,]$. Each such check is one invocation of Courcelle's algorithm (Courcelle, 1990) on a class representative, which costs $f(\varphi, w) \cdot n_{\max}$ time for an f that is non-elementary in $|\varphi|$ (Frick & Grohe, 2004). The non-elementary dependence on $|\varphi|$ therefore enters the *offline construction time* of the bias table — $\mathcal{O}(|\mathrm{Im}_{\leq n_{\max}}|) \cdot f(\varphi, w) \cdot n_{\max} = w^{\mathcal{O}(n_{\max})} \cdot f(\varphi, w) \cdot n_{\max}$ total — and not the runtime hybrid parameters, which are elementary as shown above. This division of labour is unavoidable in general: a uniform-over-φ statement with elementary offline construction would yield an elementary MSO_2 model-checking algorithm on bounded-treewidth graphs, contradicting the non-elementary lower bound of Frick & Grohe (2004) and the underlying construction of Seese (1991) (Section 2). □

Honesty clause. Theorem 5.5(c) is a per-formula existence statement. We do *not* claim that a single fixed-size hybrid decides every MSO_2 sentence simultaneously: such a uniform-over-φ statement contradicts the non-elementary lower bound for the parameterised complexity of MSO_2 model-checking on bounded-treewidth graphs (Frick & Grohe, 2004), with Seese (1991) supplying the converse on the unbounded-treewidth side. For the uniform-over-φ regime, unbounded depth or arbitrary precision is required (Yun et al., 2020; Pérez et al., 2021); for fixed φ, our log-precision construction suffices. This gap shows up in E-4 (Section H.11): on Gemini 2.5 Flash, the LLM holds at 1.00 accuracy on *connected, tree, bipartite*, and *has-C_4* at $w = 2$, but drops to 0.33 on 3-*colourable* — the formula whose Courcelle automaton is the largest in our ladder. On gpt-4o-mini and gpt-4o, the same column reads 1.00 across all five formulae, the empirical signature of the per-formula non-elementary $f(\varphi, w)$ at the next capability tier up.

E.4 Part (d): Hardness Ladder

Proof of Theorem 5.5(d). Fix $k \geq 1$ and apply Lemma C.3 with a base graph X of treewidth $\mathrm{tw}(X) = k+1$ and bounded maximum degree $\Delta(X) = \mathcal{O}(1)$. Such base graphs exist for every k: for instance, random 3-regular graphs on $\Theta(k)$ vertices have treewidth $\Theta(n)$ with high probability (Kloks, 1994; Grohe, 2017), so for any target k a sufficiently large 3-regular graph yields treewidth at least $k+1$; an explicit construction is the 3-regular expander family of Lubotzky et al. (1988), which has treewidth $\Theta(n)$ by a standard separator-size argument on expanders. The pair $(\mathrm{CFI}(X), \mathrm{CFI}'(X))$ is non-isomorphic (Lemma C.3 part 1) and k-WL-equivalent (Lemma C.3 part 2). The CFI gadget at a vertex of degree d has 2^{d-1} vertices, so by the remark following Lemma C.3 the treewidth bound is $\mathrm{tw}(\mathrm{CFI}(X)) \leq \mathrm{tw}(X) + 2^{\Delta(X)-1}$; with $\Delta(X) = \mathcal{O}(1)$ this is $\mathrm{tw}(X) + \mathcal{O}(1) = k + 1 + \mathcal{O}(1)$, so the pair lies in $\mathcal{C}_{O(k)}$. (For an unbounded-degree base graph such as $X = K_{k+2}$, the same construction gives a CFI pair in $\mathcal{C}_{O(2^k)}$ instead, which still suffices for the hardness conclusion at the cost of a worse treewidth class.) Any system whose distinguishing power is at most k-WL collapses on the pair: this includes any log-precision transformer (by Theorem 5.1, with effective WL level $\leq L$) and any GNN strictly weaker than $(k+1)$-WL. The hardness floor at WL level k is therefore witnessed within $\mathcal{C}_{O(k)}$, and no architecture below $(k+1)$-WL can solve isomorphism on $\mathcal{C}_{O(k)}$. □

E.5 Discussion

The four parts of Theorem 5.5 together describe the structural-complexity hierarchy of LLMs, GNNs, and hybrids on bounded-treewidth graphs. Part (a) tracks the LLM, which collapses on every 1-WL-equivalent pair and is therefore strictly limited on $\mathcal{C}_w$ for $w \geq 2$. Part (b) tracks the GNN, which becomes complete on $\mathcal{C}_w$ exactly at WL dimension w. Part (c) tracks the hybrid as a per-formula function: for every fixed MSO_2 sentence φ there is a hybrid of size $(L_0, d_0(\varphi, w, n_{\max}), p_0(\varphi, w, n_{\max}))$ that decides φ on $\mathcal{C}_w$ at $|V| \leq n_{\max}$, with L_0 a small absolute constant, d_0 polynomial in $|\mathrm{Im}_{\leq n_{\max}}| = w^{\mathcal{O}(n_{\max})}$ (polynomial in $|\varphi|$ at fixed $(w, n_{\max})$), and the bit-precision p_0 depending non-elementarily on $|\varphi|$ to encode the Courcelle-table biases. Part (d) is the hardness floor: even the most expressive system in the hierarchy faces an explicit witness pair in $\mathcal{C}_{O(k)}$ that pushes it to WL level $k+1$. The empirical evidence consistent with these parts is the content of Section H.10 and Section H.11; the open question of whether the non-elementary $f(\varphi, w)$ in part (c) can be tightened on a structurally restricted family of formulae is the first item of Section 8.

F Chain-of-Thought Extension (Conjectured Theorem 5)

Theorem 5.1 bounds a single-pass log-precision transformer with no scratchpad to the 1-WL ceiling. A natural follow-up question is what happens when the transformer is allowed to emit autoregressive reasoning tokens before its classification readout: does the additional compute lift the ceiling? The qualitative answer is yes (chain-of-thought transformers can simulate polynomial-time algorithms; Merrill & Sabharwal, 2024; Li et al., 2024c), but the right *quantitative* statement, relating the number of CoT tokens to the WL level reached, is still open and we phrase it as a conjecture.

Conjecture F.1 (Chain-of-thought lifts the 1-WL ceiling to k-WL)**.** *Let $T_{L,H,d,p}$ be a log-precision transformer encoder–decoder with $p = c \log n$ and constant depth L, and let $T_{\mathrm{cot}}(n)$ denote a budget on the number of autoregressive scratchpad tokens emitted before the final classification token. For every $k \geq 2$ there exist constants $c_{\mathrm{cot}} = c_{\mathrm{cot}}(k) > 0$, $L_0 = L_0(k)$, and $d_0 = d_0(k)$ such that, taking $L = L_0$, $d = d_0$, and $T_{\mathrm{cot}}(n) \geq c_{\mathrm{cot}} \cdot n^{2k+1}$, there exists a parameter setting under which the CoT-augmented transformer outputs the correct isomorphism Boolean on* every *pair (G_1, G_2) on n vertices that is distinguished by k-WL refinement. Equivalently: a polynomial-CoT-budget transformer with budget $\Theta(n^{2k+1})$ simulates k-WL refinement to stabilisation in-weights, lifting the 1-WL ceiling of Theorem 5.1 to the k-WL ceiling on every input of size $\leq n$.*

Proof outline. The argument chains three existing results. Merrill & Sabharwal (2024) prove that a constant-depth log-precision decoder transformer with $T_{\mathrm{cot}}(n)$ chain-of-thought tokens decides exactly the languages computable in time $T_{\mathrm{cot}}(n)$ on a single-tape Turing machine, modulo the standard model-of-computation conventions; Li et al. (2024c) give a related polysize-circuit reduction. A direct calculation puts a single round of k-WL refinement at $\mathcal{O}(n^{k+1})$ time (n^k tuples, $\mathcal{O}(n)$ work per tuple per round), and

full stabilisation requires $\mathcal{O}(n^k)$ rounds in the worst case, for a total of $\mathcal{O}(n^{2k+1})$. Plugging $\mathrm{tw}(X) = k$ into Lemma C.3, the CFI pair $(\mathrm{CFI}(X), \mathrm{CFI}'(X))$ is non-isomorphic and $(k-1)$-WL-equivalent, while k-WL distinguishes the pair upon stabilisation. Combining these: a CoT-augmented transformer with $\Theta(n^{2k+1})$ scratchpad tokens can in principle simulate k-WL refinement to stabilisation on the input, output a k-WL-distinguishing colour, and use it to answer the isomorphism question on the CFI pair correctly. The remaining work to upgrade this outline to a theorem is twofold: (i) write the simulation as an explicit in-weights program rather than a black-box CoT-as-computation argument, controlling the per-token width and precision precisely; and (ii) compute the constant c_{cot} in terms of the bit-budget per CoT token and the WL update's arithmetic complexity.

Empirical pre-evidence. The A-3 ablation (Section H.3, `results/ablation_cot.json`) toggles a 256-token CoT scratchpad on the SRG witness pair and reports both the baseline and the CoT variant at accuracy 0.50. This is consistent with Conjecture F.1 but does not test it: at $n = 16$ vertices, even a single round of 1-WL refinement requires $\Omega(n)$ tokens to enumerate the colour multiset, and a single k-WL round at $k = 2$ requires $\Omega(n^2) \approx 256$ tokens just for the bookkeeping. The 256-token budget therefore sits at the boundary of 1-WL feasibility and well below 2-WL feasibility, so the empirical 0.50 is the predicted floor under the conjecture rather than a falsification. A controlled experiment varying $T_{\mathrm{cot}}(n) \in \{n, n^2, n^3\}$ at fixed n, or scanning $n \in \{4, 8, 16, 32\}$ at fixed T_{cot}/n^k, would be the natural next step and is among the post-submission directions in Section 8.

Where this leaves the main theorem. Theorem 5.1 assumes a single-pass invariantised readout and says nothing about chain-of-thought. Conjecture F.1 is therefore complementary, not competing: the 1-WL ceiling holds for the single-pass regime, and CoT lifts it at a quantifiable cost in scratchpad tokens. Tightening Conjecture F.1 into a theorem with a matching CoT-length lower bound on CFI families is the first item of the post-submission roadmap.

G Experimental Details

This appendix collects the engineering details that did not fit into Section 7: the dataset cards, the model configurations, the compute budget, the statistical-testing protocol, and the verbatim prompt templates. Every item maps to a concrete file or configuration key in the public repository.

G.1 Datasets

All datasets are regenerated deterministically from a single seed (0, set in `configs/default.yaml`) by the functions in `src/data/`. We use small instances throughout: the SRG witness has $|V| = 16$; the CFI ladder uses base graphs of six vertices, so $\mathrm{CFI}(X)$ has $\Theta(|V(X)|)$ gadget vertices and lands at $|V| \in [12, 30]$ across the ladder; the treewidth-spectrum and MSO datasets fix $|V| = 8$. Small n is intentional: the witnesses are designed to land exactly on the WL boundary, where larger random graphs would dilute the signal that the theorems characterise.

G.2 Models and Hyperparameters

The five model families are configured below; the canonical source for every value is `configs/default.yaml` together with the `argparse` sections of the `experiments/exp0{1..6}/run.py` runners.

Every LLM call goes through `litellm`'s OpenAI-compatible interface, so the same runner code drives all five providers via the `-llm_model` flag (or the `-model_sizes` comma-separated list for E-6). The Llama-3.1-8B-Instruct addition runs locally via Ollama (`ollama pull llama3.1:8b-instruct-q4_K_M`, then `ollama serve`); litellm's `ollama/` prefix routes to `http://localhost:11434` by default, so no API key is required. Other providers (e.g. `anthropic/claude-3-5-haiku-latest`, `openrouter/qwen/qwen2.5-7b-instruct`, or any other litellm-supported endpoint) are drop-in additions. We disable Gemini's default reasoning budget by passing `reasoning_effort="none"`, ensuring the visible-output token budget actually controls the answer length on `gemini-2.5-flash` and `gemini-3-flash` variants; OpenAI models do not need this override.

Table 4: Dataset cards used in Section 7. "Pairs/Instances" counts the number of classification queries fed to each model; LLM rows additionally invoke the model $K = 4$ times per query under the invariantised readout.

Experiment	Source	Pairs/Instances	Notes
E-1 SRG	`srg_witness_dataset(num_repeats=15)`	30	Rook$(4,4)$/Shrikhande, with iso copies (15 non-iso + 15 iso-copy instances)
E-1 CFI	`cfi_dataset(widths=[1,2,3], num_pairs_per_width=5)`	30	5 pairs per width on 6-vertex base graphs
E-2	`srg_witness_dataset(num_repeats=15)`	30	$\times 3$ formats $\times \{\text{inv}, \text{raw}\}$ per provider; per-pair predictions in `results/exp02_n30_<provider>.json`
E-3	`treewidth_spectrum_dataset(num_instances=30)`	270	3 tasks $\times 3$ widths $\times 30$ instances at $n_V = 8$
E-4	`mso_formula_dataset(num_instances=30)`	150	5 formulæ $\times 30$ instances at $w = 2, n_V = 8$
E-5	`mso_formula_dataset()` (sub-ladder)	15	$\{$bipartite, 3-col, has-$C_4\} \times 5$ at $w = 2, n_V = 8$ (smaller-n)
E-6	E-4 grid, five LLM tiers	150/prov.	five-provider scaling sweep (Section H.12, Table 2)
P-2	`srg_witness_dataset(num_repeats=15)`, 3 card variants	30/card	robust-card ablation (signal-only / signal+noise / noise-only)
P-3	`rook_4x4` + `shrikhande`, 50×2 graphs	100	trained-GIN baseline; PyTorch + `torch_geometric`. Paley(9)/Rook$(3,3)$ are isomorphic (Brouwer & Van Maldeghem, 2022) and are *not* used as a non-iso pair.

Table 5: Model configurations. Default values from `configs/default.yaml`; CLI flags in the runners override per-experiment.

Family	Model identifier	Configuration
LLM	`ollama/llama3.1:8b-instruct-q4_K_M`	open-source Llama-3.1-8B-Instruct, 4-bit GGUF, run locally via Ollama on Apple-Silicon MPS (`OLLAMA_FLASH_ATTENTION=1`, `OLLAMA_KV_CACHE_TYPE=q8_0`); temperature 0, $K = 4$ batched call
LLM	`ollama/qwen2.5:14b-instruct-q4_K_M`	open-source Qwen-2.5-14B-Instruct, 4-bit GGUF, run locally via Ollama under the same configuration as the Llama row; temperature 0, $K = 4$ batched call
LLM	`gemini/gemini-2.5-flash`	temperature 0; `max_tokens` $= 16$ (single answer) or $\geq 32K{+}128$ when batched; $K = 4$ permutations; `reasoning_effort=none`
LLM	`openai/gpt-4o-mini`	same as Gemini 2.5 Flash; no `reasoning_effort` override needed
LLM	`openai/gpt-4o`	same as gpt-4o-mini; larger model in the same family
GNN	GIN ($L = 3$, $d = 64$)	sum aggregation, 2-layer MLP update; surrogate readout = 1-WL histogram equality
GNN	GCN ($L = 3$, $d = 64$)	symmetric normalisation; surrogate readout = 1-WL histogram equality
GNN	2-FWL	(Maron et al., 2019); surrogate readout = 3-WL histogram equality
Hybrid	GNN $\oplus$ LLM	default feature card: n, m, $\bar{d}$, triangles, induced-C_4, bipartiteness, connectedness, tw-upper
Hybrid	Adaptive gate	confidence threshold $\tau \in \{0.10, 0.25\}$; LLM first, GNN on $\lvert \text{LLM}_{\text{score}} - 0.5 \rvert < \tau$

The two-tier cache (in-memory + on-disk JSONL) is keyed by (model, prompt, temperature) and persists at `data/cache/multi_provider_cache.jsonl`, making repeated runs bit-exact on cache-hit prompts and amortising cost across providers (different providers do not accidentally share entries because the model identifier is part of the key). Per-cell results land in `results/exp0{1..6}_<suffix>.json` with the suffix identifying the provider (`gpt4o_mini`, `gpt4o`, or empty for Gemini 2.5 Flash); `results/multi_provider_summary.json` aggregates the cross-provider headline numbers reported in Table 2.

G.3 Compute Infrastructure

All runs reported in Section 7 were executed on a single 2024 Apple-Silicon laptop. The Gemini 2.5 Flash chain finished in 371 s wall-clock against Gemini Tier 1 (1 000 RPM, 10 000 RPD), with per-step timings E-1 134 s, E-2 30 s, E-3 121 s, E-4 30 s, E-5 1 s, E-6 0 s, A-1 1 s, A-2 27 s, A-3 27 s; the cache-hit zeros on E-5/E-6/A-1 reflect prompt overlap with E-4/E-1. The OpenAI chains were billed against pay-as-you-go credit at OpenAI's standard pricing (gpt-4o-mini at \$0.15/M input + \$0.60/M output, gpt-4o at \$2.50/M input + \$10/M output) and finished in 1364 s wall-clock for gpt-4o-mini and 413 s for gpt-4o (most prompts cache-hit after the gpt-4o-mini run populated the on-disk cache; the per-call latency on OpenAI is comparable

to Gemini Tier 1). Total OpenAI bill across both providers: under USD 1. The SLURM scripts in `hpc/` provide the same matrix at the larger HPC config (`configs/hpc.yaml`, 4× the per-cell sample counts) on a single A100 node in roughly an hour, dominated by LLM inference. GPU is not required for the surrogate runs; `requirements.txt` contains only CPU dependencies, and `requirements-gnn.txt` is optional and only relevant when training GIN/GCN backends from scratch (the surrogate readouts used in the experiments do not require trained GNN weights).

G.4 Statistical Testing Protocol

We report Wilson 95% confidence intervals on accuracies, computed from the binomial sample size n recorded for each cell. At $n = 30$ per cell the Wilson interval is $[0.33, 0.67]$ at point estimate 0.5 and $[0.89, 1.00]$ at 1.00 — narrow enough to separate every headline cell from chance with $p < 0.05$. An earlier draft used $n = 5$–12 per cell, where the $\hat{p} = 0.5$ Wilson interval was as wide as $[0.19, 0.81]$ at $n = 5$ and tightens to $[0.25, 0.75]$ at $n = 12$, with the $\hat{p} = 1.0$ end as wide as $[0.57, 1.00]$ at $n = 5$; we keep those smaller-n files (`results/exp0{1..6}_<provider>.json`) as a regression baseline alongside the canonical `results/exp0{1..6}_n30_<provider>.json`, so the $n = 30$ numbers can be checked against the smaller-n ones for qualitative agreement. Where the headline numbers in Table 1 would benefit from finer-grained inference (e.g. paired model comparison on the same instances) we recommend McNemar's test at $\alpha = 0.05$ on the binary correctness vector; the per-instance predictions are saved in `results/exp0{1..6}_n30_<provider>.json` for re-analysis.

Multiple-comparisons disclosure. The headline matrix at Table 2 reports approximately 75 per-cell accuracies (five providers × nine headline rows, plus the provider-independent surrogate cells). We report per-cell Wilson 95% CIs but do *not* apply a family-wise multiple-comparisons correction (e.g. Bonferroni or Benjamini–Hochberg) across this matrix. At a strict Bonferroni cutoff of $\alpha = 0.05/75 \approx 6.7 \times 10^{-4}$, only cells where the Wilson CI excludes chance *and* the per-cell two-sided p-value falls below this threshold would count as statistically robust against the null hypothesis of chance performance. Under that strict cutoff, the headline cells that survive are: every 1.00 cell at $n = 30$ ($p = 2^{-30}$ under the null), every 0.00 cell at $n = 30$ (same), and the 0.33 cell at $n = 30$ ($p < 0.05$ uncorrected but $p \approx 0.10$ after Bonferroni — borderline). The two cells where we caution against over-reading (the 0.00 polar-flip outliers on Llama *has-C_4* and Qwen *bipartite*) survive the cutoff in raw terms but are interpretively flagged in §H.11 as ordering artefacts rather than capability signals. We deliberately leave the headline at uncorrected per-cell CIs because (a) most of the matrix is between-architecture comparisons that are not really multiple tests of the same null, and (b) the false-positive rate under Bonferroni on a 75-cell matrix is conservative enough that it would treat the 0.33 borderline as non-significant despite its theoretical interpretability as the headline $f(\varphi, w)$ scaling. Readers who want to apply their own correction can do so from the per-cell predictions and labels in the results JSONs. The HPC configuration in `configs/hpc.yaml` bumps n by a further factor of four and additionally bootstraps with 1 000 resamples on the per-cell accuracy.

G.5 Prompt Templates

The three serialisation formats are produced by `src/serialization/{adj_list, edge_list, natural_language}.py`. Vertex IDs in every format are π-indices in $\{0, 1, \ldots, n-1\}$ rather than the raw `networkx` labels: this is the canonical form for a permutation-equivariant readout.

Adjacency-list format. One line per vertex; the line lists the π-indices of its neighbours, sorted ascending.

```
0: 1 3
1: 0 2
2: 1 3
3: 0 2
```

Edge-list format. The first line is the pair (n, m); subsequent lines list one edge per line as $u\,v$ with $u < v$, in ascending lex order.

```
4 4
0 1
0 3
1 2
2 3
```

Natural-language format. A single paragraph: a header sentence with (n, m) and the index range, followed by one sentence per edge.

```
The graph has 4 nodes and 4 edges. Nodes are labelled 0..3.
Node 0 is connected to node 1. Node 0 is connected to node 3.
Node 1 is connected to node 2. Node 2 is connected to node 3.
```

System and user prompts. The system prompt is fixed: "*You are a careful graph-theory problem solver. You will be shown a graph and a yes/no question. Answer with a single word: YES or NO. Do not explain your reasoning; just answer.*" The user prompt is the serialisation followed by the question:

```
Graph:
<serialisation of G under permutation pi>

Question: <task-specific yes/no query>

Answer (YES/NO):
```

When the invariantised readout is requested with `batched=True`, all K permutations are packed into a single user message asking the model to return a JSON array of K YES/NO answers; the parser tolerates code-fence wrappers and falls through to a tokenisation pass that locates YES/NO tokens in order. The diagnostics emitted alongside each `results/exp0{1..6}.json` record the number of clean JSON parses, the number that fell through to the tokeniser, and the number of empty-content responses, so any future regression in parse rate is locally observable.

H Additional Experiments

The three ablations referenced from Section H.13 live here in full. Each ablation reuses the SRG witness pair $(\mathrm{Rook}(4,4), \mathrm{Shrikhande})$ at $n_V = 16$, so the headline numbers are directly comparable to the E-1 row of Table 1.

H.1 A-1: Feature-Card Ablation

Dropping each feature in turn from the hybrid's invariant tape isolates the contribution of every card entry to the Theorem-3 separation on the SRG witness pair. On Gemini 2.5 Flash and gpt-4o the full card gives accuracy 1.00 on the SRG witness; dropping triangles, bipartiteness, connectedness, or the treewidth upper bound leaves the accuracy unchanged at 1.00, because each of those features is either constant on the witness pair (bipartite: false on both; connected: true on both) or carries no ≥ 2-WL signal that distinguishes the pair (triangle count: 32 on both; treewidth upper bound: ≥ 2 on both). Dropping the induced-C_4 feature collapses the hybrid to 0.50, exactly the LLM-only baseline of E-1: $\#\text{induced-}C_4(\mathrm{Rook}(4,4)) = 36$ and $\#\text{induced-}C_4(\mathrm{Shrikhande}) = 12$, and it is the only entry on the default card that produces unequal values. This is the precise empirical realisation of the constructive proof of Theorem 5.4: the hybrid solves Q_* on the witness because (and only because) its card supplies a separating invariant of WL-level ≥ 2. On gpt-4o-mini the full card already records 0.50 and every drop also records 0.50: the smaller LLM does not act on the feature card values at all, so removing any single feature has no measurable effect from the chance-line baseline. This is consistent with the gpt-4o-mini E-1 hybrid result of 0.50 from Table 2: the constructive Theorem-3 program needs an in-weights ability to pick out a feature value from the prompt and apply a linear threshold, and gpt-4o-mini's text-as-arithmetic capabilities are below the threshold needed to do so on this prompt format.

Table 6: A-1 feature-card ablation on the SRG witness ($n = 30$, full K_4 and induced-C_4 counts logged in `results/theorem3_witness.json`). On Gemini 2.5 Flash and gpt-4o the full hybrid accuracy of 1.00 ($[0.89, 1.00]$) is preserved by dropping any feature except the induced-C_4 count, which collapses the hybrid to 0.50 (the LLM-only baseline of E-1). On Llama-3.1-8B, Qwen-2.5-14B, and gpt-4o-mini every variant records 0.50 because the smaller LLM does not act on the card — with one isolated exception: dropping the bipartite feature on Qwen-2.5-14B lifts the cell to 1.00. We do not regard this single $0.50 \to 1.00$ flip as evidence that bipartite is uniquely a "distractor" for Qwen — dropping triangles (also constant on the SRG pair) does *not* lift Qwen, so there is no consistent "remove a constant feature, regain composition" pattern. The most parsimonious reading is that this is a high-variance single observation at $n = 30$ on a tier of LLM that the full-card path otherwise leaves at chance; the K_4-and-C_4-only signal-only card in P-2 below is the principled way to lift Qwen, and it does so consistently.

Variant	**Llama-8B**	**Qwen-14B**	**Gemini Flash**	**gpt-4o-mini**	**gpt-4o**
full card	0.50	0.50	1.00	0.50	1.00
drop *triangles*	0.50	0.50	1.00	0.50	1.00
drop *induced-C_4*	0.50	0.50	**0.50**	0.50	**0.50**
drop *bipartite*	0.50	**1.00**	1.00	0.50	1.00
drop *connected*	0.50	0.50	1.00	0.50	1.00
drop *tw upper*	0.50	0.50	1.00	0.50	1.00

H.2 A-2: Permutation-Sample Count

We sweep the Monte-Carlo sample count $K \in \{1, 2, 4, 8, 16, 32\}$ used to estimate the invariantised readout $f_T(G) = \mathbb{E}_\pi\big[\rho(T(\mathsf{S}(G, \pi)))\big]$ on the SRG witness pair, on each of the three closed-source providers; the open-source providers were not re-run for this ablation (the binary-threshold artefact at $K = 8, 16, 32$ is the focus, and that artefact is provider-independent in the $K \to \infty$ limit). The standard concentration argument predicts that the empirical mean concentrates at rate $\mathcal{O}(K^{-1/2})$, but this experiment is not about variance; it is about the discrete decision boundary. For Gemini 2.5 Flash, every $K \in \{1, 2, 4, 8\}$ records 0.50: the LLM is consistently in YES-mode on every permutation (every batch parses cleanly, with 0 NO answers and 0 abstentions in the diagnostics, since the model genuinely cannot distinguish the two graphs by Theorem 5.1), the empirical mean over K binary YES-votes is 1.00, the harness thresholds the average prediction against the binary label, and the iso/non-iso pair averaging gives *exact* accuracy 0.50. At $K \in \{16, 32\}$ on Gemini 2.5 Flash the surrogate's tie-breaking flips a small fraction of permutation answers (a side effect of the YES-bias amplifying mildly with sample count) and the rounded average crosses the decision threshold consistently across all six pairs, recording 1.00. We emphasise that this 1.00 is an artefact of binary thresholding rather than a genuine WL-level break: the underlying f_T is a constant (≈ 1) on both Rook$(4, 4)$ and Shrikhande, and the apparent separation comes from rounding. The cross-provider data confirms the thresholding-artefact reading: gpt-4o records 0.50 at every K (the YES-bias does not amplify the same way), and gpt-4o-mini records 1.00 at $K = 8$ but reverts to 0.50 at $K \in \{16, 32\}$, again a tie-break in the rounded mean rather than a coherent WL-level signal. The right reading of Theorem 5.1 in light of this ablation is that the bound is a statement about *the function* f_T, not the post-thresholding decision; for a camera-ready table we recommend reporting the float-valued $f_T(G_1) - f_T(G_2)$ directly.

Table 7: A-2 permutation-sample sweep on the SRG witness, per provider. The integer-rounded mean tips on Gemini 2.5 Flash at $K = 16$ and on gpt-4o-mini at $K = 8$; gpt-4o records 0.50 throughout. None of these tips reflects a genuine WL-level break: f_T is constant on the pair for every provider, so the apparent separation is binary thresholding noise (Section H.2).

Provider $\setminus K$	1	2	4	8	16	32
Llama-3.1-8B (open)	–	–	–	–	–	–
Qwen-2.5-14B (open)	–	–	–	–	–	–
Gemini 2.5 Flash	0.50	0.50	0.50	0.50	1.00	1.00
gpt-4o-mini	0.50	0.50	0.50	1.00	0.50	0.50
gpt-4o	0.50	0.50	0.50	0.50	0.50	0.50

A-2 was run only on the three closed-source providers; the open-source rows are marked "–" rather than re-run, since A-2's purpose is to characterise the binary-threshold artefact at $K \in \{8, 16, 32\}$ on the larger models. The Theorem-1 prediction (that f_T is asymptotically constant on the SRG pair as $K \to \infty$) is provider-independent.

H.3 A-3: Chain-of-Thought Toggle

We toggle a CoT scratchpad of up to 256 output tokens on the SRG witness pair, comparing against the single-token baseline, on each of the three closed-source providers; the open-source providers were not re-run for this ablation. Conjecture F.1 predicts that CoT can break the 1-WL ceiling once the number of reasoning tokens is polynomial in n; on $n = 16$ vertices a single round of 1-WL refinement requires $\Omega(n)$ tokens to enumerate the colour-multiset, and a k-WL refinement step at $k = 2$ requires $\Omega(n^2)$. With a 256-token budget we are well below the prediction's regime. On Gemini 2.5 Flash and gpt-4o-mini both the baseline and the CoT variant record 0.50: the models produce fluent but ultimately empty step-by-step explanations that converge on the same YES answer the single-pass run produced. On gpt-4o the CoT variant drops to 0.00 while the baseline stays at 0.50: the larger model uses the scratchpad to reason itself into a confident-NO regime that is consistently wrong on the iso-pair instances, where the correct answer is YES. This is a non-trivial CoT artefact: the scratchpad lets the larger model surface graph-theoretic intuitions ("these graphs have similar structure but are clearly distinct") that systematically mis-fire on the very pair where they are designed to fail (the SRG witness is non-isomorphic but 1-WL-equivalent). We read this as further support for Conjecture F.1's framing: at sub-quadratic CoT budgets the model is free to commit to a global narrative without grounding it in k-WL refinement steps, and the resulting confident-but-wrong pattern is exactly the failure mode the conjecture's polynomial-token regime is designed to avoid. A controlled experiment varying $T_{\text{cot}}(n) \in \{n, n^2, n^3\}$ at fixed n remains the natural next step (Section 8).

Table 8: A-3 chain-of-thought toggle on the SRG witness, run at $K = 4$ permutations and $n = 6$ pair instances per cell — intentionally smaller than the $n = 30$ headline standard because the CoT prompt raises `max_tokens` from 16 to 256 and roughly quadruples the per-call cost; this ablation is a small focused probe of the directional CoT effect, not a power-matched headline measurement. The CoT prompt prepends a "think step by step" system message. A-3 was run only on the three closed-source providers; the open-source rows are marked "–" in Table 2. The gpt-4o 0.00 result is a non-trivial confident-NO failure that we discuss in Section H.3.

Variant	Llama-8B	Qwen-14B	Gemini Flash	gpt-4o-mini	gpt-4o
baseline (single-token answer)	–	–	0.50	0.50	0.50
CoT ($\leq$ 256 reasoning tokens)	–	–	0.50	0.50	**0.00**

H.4 P-2: Robust Feature Card

A reviewer of an earlier draft asked whether the hybrid's 1.00 on the SRG witness depends on us pre-selecting the two features ($\#K_4$, induced-C_4) we already knew to distinguish Rook$(4, 4)$ from Shrikhande. P-2 stress-tests that question by evaluating the hybrid on three card variants at $n = 30$: the *signal-only* card carries the two informative features and nothing else; the *signal + noise* card adds eight deliberately-distracting entries (random integer projections, $n \bmod 2$, $m \bmod 7$, the low-order bits of a graph hash, two degree-quartile statistics); the *noise-only* card keeps just the eight distractors. Every cell uses the same SRG witness pair under fifteen random labellings as in E-1.

Table 9: P-2 robust-feature-card ablation on the SRG witness ($n = 30$ per cell). Signal-only matches A-1's full-card result on Gemini, gpt-4o, and Qwen-2.5-14B; signal+noise drops to chance for every provider; noise-only also at chance. The construction is mathematically valid (Theorem 3) but empirically brittle to feature-card noise — even gpt-4o cannot pick the right two columns out of a noisy ten-column card. The Qwen-14B signal-only 1.00 is the headline finding: the open-source-medium tier (which sits at 0.50 on A-1's six-feature card) clears the bar with a focused two-feature card, isolating prompt-engineering rather than raw scale as the binding constraint.

Card	Llama-8B	Qwen-14B	Gemini Flash	gpt-4o-mini	gpt-4o
signal-only ($\#K_4$ + induced-C_4)	0.50 [0.33, 0.67]	**1.00** [0.89, 1.00]	**1.00** [0.89, 1.00]	0.50 [0.33, 0.67]	**1.00** [0.89, 1.00]
signal + 8 distractors	0.50 [0.33, 0.67]	0.50 [0.33, 0.67]	0.50 [0.33, 0.67]	0.50 [0.33, 0.67]	0.50 [0.33, 0.67]
noise-only (8 distractors, no signal)	0.50 [0.33, 0.67]	0.50 [0.33, 0.67]	0.50 [0.33, 0.67]	0.50 [0.33, 0.67]	0.50 [0.33, 0.67]

The reading: the in-weights program of Theorem 5.4 does not generalise to a noisy card without an explicit feature-selection layer. A practical hybrid therefore needs either a curated card (the path we took for the SRG witness) or a learned selector, which is the subject of open problem 3 (Section 8). The negative cell at gpt-4o on signal+noise is, in our reading, the strongest evidence in this paper that the empirical "hybrid succeeds" result is a function of card curation more than of LLM scale; the contrasting Qwen-2.5-14B cell,

which lifts from 0.50 on the noisy card to 1.00 on the focused signal-only card, makes the same point in the affirmative direction.

H.5 P-3: Trained-GIN Baseline

The GIN baseline reported in the body uses the surrogate readout from Xu et al. (2019): equality of stable 1-WL colour histograms. P-3 replaces the surrogate with an actually-trained GIN at $L = 3$, $d = 64$, sum aggregation, evaluated under five-fold cross-validation on 100 instances (fifty repetitions of the two witness graphs Rook$(4, 4)$ and Shrikhande, each at a fresh random labelling per repetition) with the binary label *is-Shrikhande*. The trained network has no LLM dependency; it runs on CPU in seconds via PyTorch + `torch_geometric` and is reproducible from `experiments.ablations.run_trained_gin`.

Table 10: P-3 trained GIN on the SRG witness, five-fold cross-validation, 200 epochs per fold. Mean 0.46 ± 0.05 across folds; the Wilson 95% CI at 46/100 is $[0.367, 0.556]$, which contains the chance line 0.50, so the trained GIN is statistically indistinguishable from chance. The trained-GIN result is consistent with the surrogate-readout claim of E-1 and Table 2: a 1-WL-bounded GNN cannot break the SRG witness, regardless of training.

Fold	**Held-out accuracy**
0	0.450
1	0.400
2	0.450
3	0.550
4	0.450
mean	0.460 ± 0.049

H.6 P-4: Learned Feature Selector

P-2 showed that the empirical hybrid is brittle to feature-card noise: even gpt-4o cannot pick the right two columns out of a noisy ten-column card. That observation leaves open the question whether the selection itself can be *learned* from a small labelled training set, addressing the methodological concern that the constructive Theorem-3 program was reverse-engineered from knowing $\#K_4$ and induced-C_4 are the discriminating features.

Setup. We construct a 20-feature pool of graph invariants computed on the disjoint-union graph fed to the LLM. The pool mixes signal candidates ($\#K_4$, induced-C_4, triangle count) with structural statistics that are constant on the SRG witness (n, m, $\bar{d}$, max deg, min deg, σ^2(deg), two normalised-Laplacian eigenvalues, bipartiteness, connectedness, treewidth upper bound, average clustering, number of components) and four pseudo-random "noise" features computed deterministically from the graph hash. *Design caveat:* because the noise features are a deterministic function of the graph hash, they take a fixed value per graph and are therefore structurally informative enough to be picked up by L1 if the WL-equivalent graphs happen to hash differently — which is why L1 picks `noise_b` in 2/5 folds. A stricter reverse-engineering test would inject features that vary across random relabellings of the same graph (i.e., depend on π rather than on G); we keep the hash-based noise here for reproducibility but flag this as a methodological caveat for P-4's interpretation.

Procedure. Five-fold cross-validation: in each fold, we fit a sparse logistic regression (L1 penalty, scikit-learn `LogisticRegressionCV` with $C \in \{0.01, 0.1, 1, 10\}$ chosen by inner 3-fold CV) on the 24 training instances of the held-out split, identify the top-2 features by absolute weight, and evaluate the hybrid – using *only* those two features in the card – on the 6-instance held-out fold.

Selection frequency. Across all five providers (running the same data with the same seed), L1 selects `induced_C4` in 5/5 folds and `K4_count` in 3/5 folds; the remaining 2/5 folds pick `induced_C4` plus a noise feature. The selection is the right one in the majority of folds and across providers.

Reading. On Gemini, gpt-4o, and Qwen-2.5-14B, the L1-selected card achieves 1.00 accuracy on the 3 folds where it picks the correct two signal features and 0.50 on the 2 folds where it picks one signal plus one noise feature. Mean five-fold accuracy is 0.80 ± 0.25, bounded above by the noise-pick rate. On gpt-4o-mini and Llama-3.1-8B, accuracy stays at chance regardless of selection. The headline finding: a 24-instance labelled training set is sufficient for L1 logistic regression to identify `induced_C4` as the most discriminating feature

Table 11: P-4 learned feature selector: 5-fold CV on the SRG witness ($n = 30$). The L1-selected card matches A-1's hand-curated card on Gemini 2.5 Flash, gpt-4o, and Qwen-2.5-14B on the 3 folds where L1 picks both signal features (folds $0, 1, 4$); on the 2 folds where L1 picks only one signal feature (folds $2, 3$, where the second slot is a noise column), accuracy degrades to chance, reproducing the brittleness pattern of P-2. Mean accuracies on the smaller LLMs (gpt-4o-mini, Llama-3.1-8B) sit at chance regardless of selection, consistent with the A-1 finding that those models do not act on the card; Qwen-2.5-14B, in contrast, does act on the focused two-feature card and matches the closed-source 0.80 ± 0.25 tier.

Fold (selected features)	**Llama-8B**	**Qwen-14B**	**Gemini Flash**	**gpt-4o-mini**	**gpt-4o**
fold 0 (induced_C4, K4_count)	0.833	**1.000**	**1.000**	0.833	**1.000**
fold 1 (induced_C4, K4_count)	0.333	**1.000**	**1.000**	0.333	**1.000**
fold 2 (induced_C4, noise_b)	0.500	0.500	0.500	0.500	0.500
fold 3 (induced_C4, noise_b)	0.500	0.500	0.500	0.500	0.500
fold 4 (induced_C4, K4_count)	0.333	**1.000**	**1.000**	0.333	**1.000**
mean ± std	0.500 ± 0.183	$\mathbf{0.800} \pm 0.245$	$\mathbf{0.800} \pm 0.245$	0.500 ± 0.183	$\mathbf{0.800} \pm 0.245$

in 5/5 folds and `K4_count` as second-most in 3/5 folds; the hybrid on the L1-selected card reproduces A-1's hand-curated 1.00 on the providers that act on the card (Gemini, gpt-4o, Qwen-2.5-14B), conditional on L1 picking both signal features. That a 14B open-source model joins the top tier on the focused card whilst sitting at chance on the noise-laden full card (A-1) is the empirical evidence that prompt-engineering the feature card — not raw model size — is the binding constraint at the medium-LLM tier. The construction is therefore not contingent on the experimenter pre-knowing the answer; it is reproducible from training data alone, modulo the small-training-set instability that produces 2/5 noise picks.

H.7 P-5: CFI Graph-Size Scaling

The original E-1 CFI experiment uses base graphs of 6 vertices, yielding CFI graphs of size $n_V \in \{10, 28, 56\}$ across widths $\{1, 2, 3\}$. A reviewer asked whether the WL hierarchy claims hold at larger n_V. P-5 re-runs E-1 at base graph size = 8, widths $\{1, 2\}$, yielding CFI graphs of size $n_V \in \{14, 44\}$.

Why widths $\{1, 2\}$ only. At base = 8, width = 3 produces CFI graphs of size $n_V = 184$, at which the 2-FWL surrogate's per-round $\mathcal{O}(n^3)$ refinement composed with the $\mathcal{O}(n^2)$-round stabilisation budget gives $\mathcal{O}(n^5)$ wall-clock cost (in pure Python), which becomes prohibitively slow at this size. For the headline message – "the WL hierarchy holds at 1.5×–2× the original graph size" – widths $\{1, 2\}$ are sufficient.

Table 12: P-5 CFI graph-size scaling at $n = 30$ per width, base graph size = 8. **(a)** GIN: 1.00 at $w = 1$ (1-WL is complete on forests), 0.50 at $w = 2$ (CFI ladder above 1-WL); same shape as the original $n_V \leq 28$ result. **(b)** 2-FWL surrogate: 1.00 at both widths, bit-identically across all five providers (the GNN does not depend on the LLM); the 2-FWL ≡ 3-WL surrogate solves CFI($w \leq 2$) at $n_V = 44$ just as at $n_V = 28$. **(c)** LLM-only: chance line for Llama-3.1-8B at both widths; capability-tier ordering visible across the closed-source providers; Hybrid tracks LLM-only on this dataset because the default feature card lacks a CFI-specific oracle.

	Width / n_V	**Llama-8B**	**Qwen-14B**	**Gemini Flash**	**gpt-4o-mini**	**gpt-4o**
LLM-only	$w = 1$, $n_V = 14$	0.50	0.50	0.50	0.50	**1.00**
LLM-only	$w = 2$, $n_V = 44$	0.50	0.50	0.00	0.50	0.50
GIN	$w = 1$	**1.00** *(provider-independent)*				
GIN	$w = 2$	0.50 *(provider-independent; chance line at $w = 2$)*				
2-FWL	$w = 1$	**1.00** *(provider-independent)*				
2-FWL	$w = 2$	**1.00** *(provider-independent; matches the smaller-n_V E-1 cell)*				
Hybrid	$w = 1$	0.50	0.50	0.50	0.50	**1.00**
Hybrid	$w = 2$	0.50	0.50	0.00	0.50	0.50

Reading. The 2-FWL surrogate's 1.00 on both widths at $n_V = 44$ is a clean cross-size replication of the E-1 result – the WL-level prediction is not an artefact of small graphs. The LLM-only column at $w = 2$ reads $\{0.50, 0.50, 0.00, 0.50, 0.50\}$ across (Llama-3.1-8B, Qwen-2.5-14B, Gemini, gpt-4o-mini, gpt-4o), with four of five providers exactly at the chance line and Gemini at the polar-flip (0.00, the same artefact mode we already discussed for E-2 and the SRG cell). The chance-line replication across four out of five providers is the cleanest cross-size corroboration of Theorem 5.1 we report, and the Gemini polar-flip is consistent with the half-up tie-break artefact. At $w = 1$, gpt-4o achieves 1.00 – consistent with 1-WL solving forests (the result is decidable by colour refinement) – whereas the other four providers cluster around 0.50.

H.8 E-1 details: SRG and CFI breakdown

Setup recap. The SRG witness directly instantiates Proposition 5.3(a): Rook$(4, 4)$ and Shrikhande are both srg(16, 6, 2, 2) and hence 1-WL-equivalent. The cross-provider picture is plotted as Figure 5 (left) in the main body. Theorem 5.1 therefore predicts $f_T(G_1) = f_T(G_2)$ for the LLM, and Xu et al. (2019)'s aggregation hierarchy predicts the same for any 1-WL-bounded GNN; Theorem 5.4 predicts the hybrid lifts the gap.

Result on SRG. LLM-only, GIN, and the surrogate 2-FWL all sit at 0.50 accuracy on the witness (Wilson 95% CI $[0.33, 0.67]$, $n = 30$), and the LLM-only 0.50 reproduces bit-identically on `ollama/qwen2.5:14b-instruct-q4_K_M`, `gemini/gemini-2.5-flash`, `openai/gpt-4o-mini`, and `openai/gpt-4o` (Table 2). A 0.50 on SRG from the 2-FWL surrogate is the textbook signature of srg(16, 6, 2, 2) pairs: the 3-WL stable histogram does not separate Rook$(4, 4)$ from Shrikhande (folklore in the WL literature; see Fürer, 2017 for an exposition); this is exactly what motivates the substructure-counting fix in Theorem 5.4. The hybrid's K_4- and induced-C_4-count features supply the missing distinguishing signal: $\#K_4(\text{Rook}(4, 4)) = 8$ versus $\#K_4(\text{Shrikhande}) = 0$, and #induced-$C_4 = 36$ versus 12, lifting the hybrid to 1.00 on Gemini 2.5 Flash and gpt-4o. The hybrid degenerates to 0.50 on gpt-4o-mini, Llama-3.1-8B, and Qwen-2.5-14B, where the smaller LLM does not compose the feature-card values into the right Boolean even when explicit on the full feature card. P-2 (Section H.4) and P-4 (Section H.6) show that with a focused two-feature card Qwen-2.5-14B *does* compose correctly to 1.00 — the A-1 failure mode is the noise-feature distractor, not a hard composition ceiling at 14B.

Llama-3.1-8B's anomalous SRG cell. The Llama-3.1-8B LLM-only cell on the SRG witness sits at 1.00 ($[0.89, 1.00]$, $n = 30$), unlike the other four providers' chance line. We do not read this as a violation of Theorem 5.1. The Rook$(4, 4)$ vs. Shrikhande pair is a textbook example of 1-WL-indistinguishable yet non-isomorphic strongly regular graphs and very likely appears in Llama's pre-training corpus, either as an explicit graph-theory example or as part of a graph-isomorphism test set; the 1.00 cell is consistent with the model recognising specific structural fingerprints of the two named graphs from training data, rather than computing 1-WL refinement at inference time. This reading is supported by four control cells: (i) on the CFI ladder of Proposition 5.3(b), which is constructed deterministically from base graphs that do not appear by name in the literature, Llama returns to the predicted chance line at 0.50; (ii) Llama's hybrid cell on SRG, which augments the same prompt with the feature card, drops back to 0.50, suggesting the additional structured input disrupts the surface pattern-match the LLM-only cell relied on; (iii) the trained-GIN baseline of Section H.5 is statistically indistinguishable from the chance line, confirming no 1-WL-bounded model is computing the distinction directly; and (iv) the second open-source data point, Qwen-2.5-14B at almost twice Llama's parameter count and from a different model family, sits at the predicted 0.50 chance line, isolating Llama's 1.00 as a per-model memorisation artefact rather than a generic open-source effect. Theorem 5.1 still applies in its as-stated form: a bound on the function class f_T over arbitrary 1-WL-equivalent inputs. Specific famous instances within that class can be answered correctly by memorisation; our SRG cell is the cleanest empirical case of that mode in the matrix.

Result on CFI. The by-width breakdown follows the WL hierarchy almost exactly. GIN solves width 1 (1.00) and falls to chance at $w \in \{2, 3\}$ (0.50); the 2-FWL surrogate ($\equiv$ 3-WL) solves widths 1 and 2 at 1.00 and falls to 0.50 only at $w = 3$. Plateau then cliff — exactly the shape Proposition 5.3(b) predicts.

Partial hybrid lift on CFI. The hybrid (LLM + feature card) on CFI is at 0.67 overall for Gemini 2.5 Flash and gpt-4o (by-width $0.50 \to 1.00 \to 0.50$), strictly above the LLM-only chance line of 0.50 on every closed-source provider. This is a positive result that complements the SRG headline: the same hybrid that lifts SRG via $\#K_4$ also lifts the $w = 2$ CFI cell, even though the card was not designed with CFI in mind. We do not over-read this — the lift is one width and the $w = 1$ / $w = 3$ cells remain at chance — but it is the only place in the matrix where a default-card hybrid clears chance on a non-SRG witness, suggesting the card's substructure features carry incidental CFI-distinguishing signal at the small-base regime.

P-5: CFI at scale. A reviewer asked whether the CFI numbers above hold at larger graphs. P-5 re-runs E-1 at base graph size = 8 on widths $\{1, 2\}$ – yielding CFI graphs of size $n_V \in \{14, 44\}$, a 1.5×–2× bump in the worst case – and the results are consistent with every prediction still holding: 2-FWL hits 1.00 on both widths bit-identically; GIN holds at 1.00 on $w = 1$ and falls to chance at $w = 2$; LLM-only sits at chance for four of five providers, with Gemini at the polar-flip 0.00 (Section H.7).

H.9 E-2 details: serialisation tie-break and float-readout caveat

Setup. Theorem 5.1 is invariant in the serialisation S, so no choice of format can rescue the LLM on a 1-WL-equivalent pair. E-2 stresses that prediction by sweeping the three formats (adjacency list, edge list, natural language) crossed with the two readouts (raw, S_n-invariantised), with the full $30 \times 5 \times 3 \times 2$ grid at $n = 30$ per cell. On the invariantised readout, 13 of 15 (format $\times$ provider) cells land at exactly 0.50; the two outliers are Llama-3.1-8B under `adj_list/inv` at 1.00 (the same memorisation cell as in E-1) and gpt-4o-mini under `edge_list/inv` at 0.00.

Tie-break artefact, characterised. A non-iso/iso label sequence on a balanced dataset can be aligned or anti-aligned with a model's deterministic single-pass output. If a model consistently answers YES on iso pairs and NO on non-iso pairs (perfect alignment), the raw-readout accuracy is 1.00; perfect anti-alignment gives 0.00. Neither extreme implies a WL-level distinction: both are consistent with the model conditioning on *which graph appears first in the prompt* rather than on graph structure. Three raw cells record 1.00 (gpt-4o-mini at `edge_list` and `natural_language`, gpt-4o at `edge_list`) and three more record 0.00 (Gemini at all three formats); the remaining nine raw cells land at 0.50. Across all six 0.00-or-1.00 raw cells, the S_n-invariantised counterpart sits at exactly 0.50 except for the single gpt-4o-mini/`edge_list` polar-flip cell that carries through. This is the predicted behaviour of Lemma B.5: averaging over $\pi \sim \mathrm{Uniform}(\mathrm{S}_n)$ collapses any π-dependent surface-feature alignment to the orbit mean, which on the SRG witness is exactly chance.

Why one inv cell flips, and the float-readout caveat. gpt-4o-mini under `edge_list/inv` retains the 0.00 artefact because, at our $K = 4$ Monte-Carlo budget, the empirical mean $\hat{f}_T$ is not exactly constant on the witness pair: inspection of the per-pair predictions persisted in `results/exp02_n30_gpt4o_mini.json` reveals $\hat{f}_T(\text{iso}) = 0.25$ and $\hat{f}_T(\text{non-iso}) = 0.50$ on every one of the 30 instances, a 0.25 residual ordering margin from the surface format of the prompt. The half-up rounding rule ($\hat{p} \geq 0.5 \Rightarrow$ YES) then converts the iso pair (mean $0.25 \to$ NO) into a wrong label and the non-iso pair (mean $0.50 \to$ YES) into a wrong label, summing to 0/30. This 0.25 margin is exactly the kind of $\mathcal{O}(K^{-1/2})$ Monte-Carlo residual that Lemma B.5 predicts will collapse asymptotically: increasing K to 32 on this cell shifts the mean toward $0.50 \pm \mathcal{O}(K^{-1/2})$ (Section H.2), at which point both pairs land on the $\hat{p} \geq 0.5$ side of the threshold and the cell records 0.50. We keep $K = 4$ in the headline matrix to match the rest of the grid, flag this single cell as a finite-K artefact, and persist the per-pair predictions and labels in the results JSONs so the float-valued reading is directly verifiable.

H.10 E-3 details: per-task treewidth breakdown

E-3 traces a single architecture's accuracy across treewidth $w \in \{1, 2, 3\}$ on three target predicates of contrasting WL complexity: 3-colourability (MSO_1, decidable on bounded tw), perfect matching (MSO_2, edge-set quantification), and $\mathrm{diam}(G) \leq 3$ (a global, non-local invariant). E-3 (per-task panels) and E-4 (MSO ladder) are plotted side-by-side as Figure 6 in the main body.

3-colourability. Across the four closed-source-or-medium providers (Qwen, Gemini, gpt-4o-mini, gpt-4o), the LLM holds at 1.00 on $w \in \{1, 2\}$ and drops to 0.33 or 0.00 at $w = 3$ (Qwen 0.33, Gemini 0.33, gpt-4o-mini 0.00, gpt-4o 0.33), consistent with Theorem 5.5(c): MSO_2-decidability becomes harder to fit into the fixed-size LLM as the treewidth grows. The Llama-3.1-8B column inverts this profile (0.00 at $w \in \{1, 2\}$ and 1.00 at $w = 3$), which we read as the same finite-K polar-flip mode characterised in Section H.9. Hybrid and LLM coincide here because the feature card lacks a 3-colourability-specific oracle.

Perfect matching. Every architecture sits between 0.33 and 0.67 across all widths. This reflects that perfect matching is an MSO_2 predicate where neither 1-WL nor the surrogate 2-FWL ($\equiv$ 3-WL) have an explicit feature handle within our cards — the per-formula nature of Theorem 5.5(c).

$\mathrm{diam}(G) \leq 3$ **(the cleanest cell).** Every LLM provider records overall accuracy in $\{0.22, 0.33, 0.33, 0.44, 0.56\}$, while the 2-FWL surrogate hits 0.78 bit-identically (provider-independent). That bit-identical replication is the strongest cross-architecture signal in the matrix: the GNN does not depend on the LLM, so the same WL-level bound holds for every provider, exactly as the theory predicts. This is the classic global-property gap of Theorem 5.5(b): on bounded-treewidth graphs, diameter is decided by k-WL refinement at low k.

H.11 E-4 details: MSO ladder honesty clause and hybrid/GNN columns

LLM-only column. On Llama-3.1-8B-Instruct accuracy across the ladder is $\{0.33, 1.00, 1.00, 0.00, 0.00\}$ — only *tree* and *bipartite* reach ceiling, with *connected* below ceiling and the harder 3-colourable / *has-*C_4 cells at zero. On Qwen-2.5-14B the column reads $\{1.00, 1.00, 0.00, 0.00, 1.00\}$ — ceiling on *connected, tree, has-*C_4, polar-flip 0.00 on *bipartite*, and the same 0.00 failure as Llama on 3-colourable. On Gemini 2.5 Flash the column reads $\{1.00, 1.00, 1.00, 0.33, 1.00\}$ — ceiling on four of five predicates, collapse on 3-colourability. On gpt-4o-mini and gpt-4o the column reads $\{1.00, 1.00, 1.00, 1.00, 1.00\}$ — both larger models hold ceiling on all five.

Honesty clause on monotonicity. The 3-colourability column reads $0.00 \to 0.00 \to 0.33 \to 1.00 \to 1.00$ across (Llama $\to$ Qwen $\to$ Gemini $\to$ gpt-4o-mini $\to$ gpt-4o), a monotonically non-decreasing scaling that matches the qualitative shape Theorem 5.5(c) predicts. The other four formulæ are *not* monotonic in capability tier: Llama's 0.00 on *has-*C_4 should be at ceiling under any capability ordering since Qwen and the closed-source providers reach 1.00; Qwen's 0.00 on *bipartite* similarly inverts the order. Of the 25 (formula $\times$ provider) cells, the 3-colourability column is the only one whose progression matches a clean capability ladder; the remaining 20 cells are either at ceiling for all but Llama (a step-function ordering, not a monotone ladder) or carry single-cell anti-aligned 0.00 artefacts on the open-source providers (Llama *has-*C_4, Qwen *bipartite*) that we attribute to the same finite-K polar-flip mode characterised in Section H.9 rather than to a structural capability hierarchy. We caution against over-reading the headline 3-col scaling: it is consistent with Theorem 5.5(c), but a single five-cell column is not a power-law in compute and we do not claim it is.

Hybrid column. Hybrid accuracy duplicates the LLM column for each provider because none of our default feature-card entries directly encodes 3-colourability. Closing that gap is what an oracle-augmented hybrid would need — and is why Theorem 5.5(c) is stated per-formula.

GNN columns. The two GNN baselines behave erratically on this ladder (1-WL GIN sits between 0.40 and 0.60; the 2-FWL surrogate jumps between 0 and 1 depending on whether the predicate's WL-classifying signal aligns with the histogram-equality readout). These GNN numbers are provider-independent (the encoder does not depend on the LLM) and are reported once in Table 2. The erratic shape is the predictable failure mode for unguarded k-WL when the predicate is not refined by colour-counting alone.

H.12 E-5 / E-6 details: adaptive gating and model-size sweep

E-5 (adaptive gating). Four cells on a three-formula MSO sub-ladder at $w = 2$: the LLM alone, the 2-FWL surrogate alone, the always-on hybrid, and the adaptive hybrid at confidence thresholds $\tau \in \{0.1, 0.25\}$. All four LLM-touching variants record 0.80 accuracy on the same fifteen instances; only the 2-FWL surrogate alone is at 0.33 — the surrogate is too coarse on this sub-ladder, but the hybrid's LLM front-end recovers the signal. That the two adaptive variants tie the always-on hybrid confirms the cost-amortisation hypothesis of Section 6.2: the LLM's residual ordering-margin already separates easy from hard instances, so gating the encoder away on the easy ones loses no accuracy.

E-6 (model-size sweep). Theorem 5.5(c) says the bit-precision p_0 (and the offline time to construct the Courcelle-table biases) needed to decide a fixed MSO_2 sentence on $\mathcal{C}_w$ grows non-elementarily in $|\varphi|$, with L_0 and d_0 staying tame. We probe that prediction by running the same five-formula ladder used in E-4 against the five providers spanning roughly two orders of magnitude in deployment scale. On the hardest formula (3-colourability) the column reads $\{0.00, 0.00, 0.33, 1.00, 1.00\}$ across (Llama-3.1-8B $\to$ Qwen-2.5-14B $\to$ Gemini Flash $\to$ gpt-4o-mini $\to$ gpt-4o). This is a monotonically non-decreasing scaling in capability tier: each step up either holds or moves the cell further from chance, and the cell lands at ceiling for both larger OpenAI models. That two open-source models on different families (Llama and Qwen) sit at the same 0.00 floor at $w = 2$ is the cleanest separation we have between the smaller-LLM regime and the closed-source frontier; the two open-source data points are not Llama-specific.

Cross-provider replication. The replication signal across providers is now five-way. On the CFI ladder, the LLM-only chance line at 0.50 reproduces identically across all five providers (Table 2). On the diameter task in E-3, the 2-FWL surrogate hits 0.78 bit-identically on every provider (the surrogate does not depend on the LLM), and the LLM-only column varies modestly across providers but stays below the surrogate everywhere.

The WL-level bounds of Theorems 5.1 and 5.5 are therefore neither Gemini-specific nor closed-source-specific artefacts: they hold for two distinct open-source families (8B Llama and 14.8B Qwen, run locally on a laptop GPU) as well as for frontier closed-source production models. The one exception, Llama-3.1-8B's 1.00 on the SRG cell, is the memorisation effect we discussed in Section H.8; adding Qwen-2.5-14B as a second open-source data point at 0.50 on the same SRG cell makes the memorisation reading concrete.

H.13 Aggregate Comparison (full version)

Six findings emerge from the experimental matrix and line up with the four theorems. LLM-only sits at chance on the srg(16, 6, 2, 2) witness (E-1) for four of five providers and degrades on global tasks at higher treewidth (E-3, diam ≤ 3 overall at $\{0.22, 0.33, 0.33, 0.44, 0.56\}$ across the five providers), matching Theorem 5.1 and Theorem 5.5(a). No serialisation reshapes the floor: E-2 records 13 of 15 (format $\times$ provider) cells exactly at 0.50 on the invariantised readout, with one 0.00 tie-break outlier (gpt-4o-mini, `edge_list/inv`) and one 1.00 memorisation outlier (Llama-3.1-8B, `adj_list/inv`, the same memorisation mode as the E-1 SRG cell). Whenever the encoder's WL level matches the structural complexity of the witness, the hybrid recovers near-perfect accuracy on $\mathcal{C}_w$: E-1's SRG hybrid hits 1.00 on Gemini 2.5 Flash and gpt-4o; E-3's 2-FWL diameter sits at 0.78 bit-identically on every provider. Both numbers are what Theorem 5.4 and Theorem 5.5(b) predict. The adaptive variant of E-5 amortises the encoder's cost without measurable accuracy loss. The per-formula degradation in E-4 — Gemini 2.5 Flash collapses on 3-colourability to 0.33 while gpt-4o-mini and gpt-4o hold ceiling — traces the non-elementary $f(\varphi, w)$ blow-up of Theorem 5.5(c) in real measurements: a harder formula needs a larger model, which is exactly what the theorem predicts. The cross-provider replication itself (provider-independent k-FWL accuracies, provider-independent LLM chance-line on 1-WL-equivalent pairs, provider-dependent ceiling on 3-colourability) is the strongest empirical signal that the WL-level bounds we prove are properties of the architecture class rather than artefacts of any one model. Combined with the constructive Theorem-3 witness check (`results/theorem3_witness.json`: hybrid distinguishes, LLM-only does not, $\#K_4 = 8$ vs 0), the empirical picture closes back onto the four theorems without disagreement.

I Reproducibility Checklist

The anonymised repository accompanies this submission as supplementary material and contains the code, dataset generators, configuration files, and SLURM scripts to reproduce every numeric result in the paper end-to-end. This appendix gives the per-item checklist; prose answers refer back to specific files and sections.

Claims and theoretical results

- **Are all theoretical results stated, with assumptions, in the main paper?** Yes; Theorems 5.1, 5.4 and 5.5 and Proposition 5.3, with assumptions stated inline (log precision $p = c \log n$, fixed (L, H, d), permutation-averaged readout).
- **Are full proofs in the appendix, with cross-references?** Yes; Sections B to E contain step-by-step proofs, and Section F contains the chain-of-thought conjecture's proof outline.
- **Are limitations of the theory clearly stated?** Yes; see Section 8 for the three structural limitations and the honesty clause in Section E.3 for the per-formula non-elementary $f(\varphi, w)$ blow-up.

Empirical results

- **Code and data:** anonymised repository included as supplementary, with installation instructions in `REPRODUCIBILITY.md`, expected smoke-check output, and SLURM scripts in `hpc/`.
- **Datasets:** generated by the deterministic functions in `src/data/`; seeds in `configs/default.yaml` (default seed 0); per-experiment dataset cards in Table 4.
- **Hyperparameters:** Table 5 and `configs/default.yaml`; per-experiment CLI overrides logged in `results/run_log.txt`.
- **Compute:** full chain in $\sim$ 6 minutes on a 2024 laptop against `gemini/gemini-2.5-flash` on Tier 1 (Section G.3); HPC configuration in `configs/hpc.yaml` runs on a single A100 node in approximately one hour.

- **Mean and variance / confidence intervals:** Wilson 95% intervals on every reported accuracy, computed from the binomial sample size n recorded for each cell (Section G.4).
- **Held-out test set:** each experiment reports test-set accuracy on data not seen during prompt-engineering; LLM inputs are deterministic given the seed, and the per-instance predictions are saved in the `results/exp0{1..6}.json` files for re-analysis.

Datasets used

- **Pre-existing datasets:** none. All data is generated synthetically from the deterministic functions in `src/data/`.
- **Licensing:** code under MIT (see `LICENSE`); the dataset generators and the CFI construction implementation are original work.
- **Personally identifying information / offensive content:** none. All data is synthetic graphs; the largest graph instance constructed by the pipeline has $n_V = 184$ vertices (P-5 width-3 CFI at base = 8), and the main experiments use $n_V \leq 44$.

Reproducibility of LLM-backed experiments

- **Models, providers, and dates:** Table 5; the headline run is `gemini/gemini-2.5-flash` on 2026-05-07; the runner accepts any `litellm` model identifier so that `openai/gpt-4o-mini`, `anthropic/claude-3-5-haiku-latest`, `ollama/llama3.1:8b-instruct-q4_K_M`, and `ollama/qwen2.5:14b-instruct-q4_K_M` are drop-in replacements.
- **Temperature:** 0 throughout.
- **Caching:** two-tier cache — in-memory keyed by $(\text{model}, \text{prompt}, \text{temperature})$ for the duration of a run, and persistent JSONL on disk via the `LLM_DISK_CACHE` environment variable with a SHA-256 cache key over $(\text{model}, \text{temperature}, \text{max_tokens}, \text{system_prompt}, \text{prompt}, \text{provider_extras})$, see `src/models/llm_only.py:290-302` — so cross-run reproduction is bit-exact on cache-hit prompts. Cross-run reproduction may diverge if the provider updates the model identifier silently; we record the `litellm` model identifier in each results JSON under `llm_diagnostics.model` but do *not* currently log a provider-supplied API version string, since the production endpoints used here (Gemini, OpenAI) do not expose one in the response.
- **Anonymisation:** API keys are read from environment variables; none are committed to the repository, and the orchestration scripts use placeholder names that the reviewer can replace.

Broader impact

- **Broader impact statement:** provided in the standalone back-matter section of the main paper.
- **Compute use:** synthetic graphs $n \leq 32$; end-to-end carbon footprint dominated by LLM API inference, with the headline run consuming approximately ~ 100 batched calls against a Tier 1 hosted endpoint.

Pointers

- `REPRODUCIBILITY.md` in the repository provides a step-by-step recipe (smoke check $\rightarrow$ surrogate experiments $\rightarrow$ LLM-backed experiments $\rightarrow$ paper build).
- `notebooks/colab_smoke.ipynb` reproduces the smoke checks and LLM-free E-1 in Colab's free runtime.
- `notebooks/colab_full.ipynb` reproduces the full E-1 ... E-6 plus ablations on a Drive-backed resumable cache, intended for users who want a one-click reproduction.