Rewiring for Computation, Not Just Accuracy: Conductance-Driven Depth Bounds for WL-Style Canonicalization

Liz Lemma Future Detective

January 18, 2026

Abstract

Recent work (including the RL-CONGEST framework) argues that WL refinement is not locally computable by message-passing GNNs without substantial depth–width capacity, and that adding virtual edges to create expander-like graphs can reduce the depth needed to simulate WL. However, existing rewiring methods in graph learning are typically justified by empirical accuracy or oversquashing heuristics, with limited formal connection to what global functions become computable at fixed depth under bandwidth limits. We develop a quantitative theory linking rewiring to provable computability speedups. We formalize a rewiring budget (edge additions) and analyze WL-style canonicalization under RL-CONGEST on the rewired graph G'. Our main results give (i) upper bounds showing that once rewiring ensures conductance $\phi(G') = \Omega(1)$ with near-linear added edges, WL refinement can be computed in polylogarithmic depth (randomized, bounded error), and (ii) lower bounds parameterized by conductance (or mixing time) showing that below this expansion regime, some canonicalization primitives require depth at least polynomial in $1/\phi$ or linear in diameter. We further propose computability-driven rewiring objectives (maximize a provable expansion proxy under a budget) and empirically validate that, holding the base model fixed, rewiring choices predictably trade accuracy for depth in recovering WL colors and solving global tasks. This work modernizes the 'virtual edges' insights from RL-CONGEST into a tight, budgeted, and actionable theory for 2026-era graph foundation models.

Table of Contents

2. 2. Background: RL-CONGEST, WL refinement as a canonicalization primitive, conductance/mixing time; relationship to expander routing/sorting and to virtual-edge results in the source material.

3. 3. Problem Setup: rewiring budget models (global R vs per-node r), target functions (exact WL refinement vs fingerprinted WL), and evaluation metrics (depth, width, failure probability).

4. 4. Upper Bounds I (Exact WL on High-Conductance Graphs): reduction from WL refinement to distributed token sorting/ranking; routing/sorting on expander-like graphs; resulting depth bounds parameterized by (w, Δ', ϕ).

5. 5. Upper Bounds II (Approximate/Fingerprinted WL): one-sided-error WL fingerprints without global perfect hashing; tradeoffs among message bits, rounds, and collision probability; when bounded-error enables polylog depth.

6. 6. Lower Bounds Parameterized by Conductance: cut-based and communication-complexity reductions showing $\Omega((1/\phi) \cdot m/(w \log n))$-type barriers for collision-free WL (and simpler canonicalization primitives) on ϕ-conductance graphs.

7. 7. Rewiring Budget vs Achievable Conductance: necessary/sufficient edges-to-add on canonical hard families (paths, grids, two-community graphs); near-tight bounds and impossibility when R is sublinear.

8. 8. Computability-Driven Rewiring Objectives: define and justify expansion proxies (e.g., resistance/Dirichlet energy surrogates) that can be estimated efficiently; propose simple rewiring heuristics with theoretical guarantees on special cases.

9. 9. Experiments (Optional but Strengthening): controlled synthetic families + real graphs; hold model fixed and vary rewiring; measure WL-color recovery, depth required, and correlation with conductance proxies; ablations on preprocessing cost.

10. 10. Discussion and Open Problems: tightness gaps (polylog factors, exact vs approximate), robustness to directed/weighted graphs, interaction with virtual nodes/tokens and shared randomness, implications for benchmark design.

1 Introduction and Motivation

Graph neural networks and, more generally, message-passing procedures are constrained not only by the expressivity of their local aggregation rules but also by the cost of disseminating information across the underlying communication graph. In the RL-CONGEST viewpoint, a depth-d message-passing architecture is a d-round distributed algorithm with a strict per-edge bandwidth constraint. In this setting, many natural primitives—including canonicalization tasks that underpin color refinement and Weisfeiler–Leman (WL) methods—are limited by the time required to coordinate globally consistent outcomes from locally formed data. We adopt the position that *rewiring* should be analyzed as a *compute accelerator*: by adding a limited number of edges in a preprocessing step, we alter the communication geometry so that the subsequent distributed computation provably completes in fewer rounds.

This perspective differs from the prevailing motivations for rewiring in representation learning. A common account, often phrased in terms of *over-squashing*, is that long-range dependencies are compressed into short messages as they traverse narrow cuts, leading to degraded predictive accuracy. While this is an important phenomenon, it entangles two effects: (i) the statistical question of what information should be transmitted, and (ii) the algorithmic question of how quickly any information can be transmitted under bandwidth constraints. Our focus is the second question. We treat the aggregation rule as fixed (a WL-type neighborhood multiset operator) and ask for a sharp, model-specific characterization of the minimal depth required to implement that operator under RL-CONGEST constraints, both with and without rewiring. In this sense, rewiring is not merely a heuristic architecture modification but an explicit resource that can be traded for round complexity.

We study one iteration of WL-1 refinement as a canonical benchmark. One WL step maps each node to a new color determined by its current color together with the multiset of neighbor colors, under perfect hashing semantics. This operation is local in its *inputs* but global in its *outputs*: in order for the new colors to be meaningful beyond a single neighborhood, equal types must be mapped to equal labels and distinct types to distinct labels in a globally consistent manner. The latter requirement forces some form of global coordination, and in bandwidth-limited models it naturally exposes the role of graph conductance (or, equivalently, mixing properties) as a bottleneck. The central hypothesis we pursue is that rewiring can be fruitfully understood as the act of increasing the conductance of the communication graph subject to explicit edge-addition budgets, thereby enabling fast global coordination primitives such as routing, sorting, and ranking.

Concretely, our model allows a preprocessing step that adds a set E_+ of edges to the given graph, producing a rewired graph G'. The post-rewiring computation proceeds by message passing on G' with per-edge bandwidth

$O(w \log n)$ bits per round and standard local computation budgets. The salient point is that we measure the depth of the *message-passing* phase, treating the rewiring as a separate resource quantified by either a global budget $R = |E_+|$ or a per-node budget r. This separation matches the practical use of rewiring in learned systems (where augmenting the adjacency can be seen as an architectural choice made prior to training/inference) and, more importantly, lets us state precise tradeoffs: how much rewiring is required to enter a regime in which WL-type refinement is computable in polylogarithmic depth, and what lower bounds persist when the rewiring fails to significantly improve expansion.

Our technical approach makes explicit a reduction that is often implicit in discussions of WL and GNNs: implementing one WL refinement step amounts to performing a distributed *canonicalization* of locally constructed keys. After one round of neighbor exchange, each node can form a representation of its WL-1 type as a key that includes its own color and the multiset of neighbor colors. The remaining task is to assign new labels so that identical keys receive the same label everywhere. This is exactly a global *ranking* problem over distributed tokens, and it is known that on high-conductance graphs, ranking can be achieved efficiently via routing and sorting primitives. Thus, in the favorable regime where G' has sufficiently large conductance, WL refinement becomes a composition of (i) local key formation and (ii) expander-style token sorting/ranking, with the overall round complexity governed by the ratio between token size and per-edge bandwidth and by the routing time guaranteed by the expansion of G'.

At the same time, we argue that this favorable regime is not automatic and cannot be obtained for free. In sparse graphs with small conductance, any collision-free relabeling that behaves like perfect hashing semantics is forced to move a large amount of distinguishing information across sparse cuts. This yields lower bounds that are inherently communication-theoretic: regardless of local computation power, an RL-CONGEST protocol cannot circumvent cut-capacity limitations. In particular, we establish families of bounded-degree graphs where conductance ϕ directly governs the minimum number of rounds needed to perform a collision-free WL refinement, scaling at least inversely with ϕ once bandwidth is fixed. This places conductance in the role of a *computational barrier*, rather than merely a correlational statistic about learning difficulty.

These observations lead to a phase-transition interpretation. When rewiring succeeds in raising conductance to a constant, the communication graph behaves like an expander for the purposes of routing, and WL-type canonicalization can be completed in polylogarithmic rounds (up to the bandwidth and degree factors induced by moving $\Theta(\Delta' \log n)$-bit keys). When conductance remains polynomially small, there exist instances where any collision-free refinement requires polynomially many rounds. The transition is not an artifact of the proof technique but a consequence of comparing expander-based upper bounds with cut-based lower bounds: the same parameter (conductance) controls both the achievable routing time and the unavoidable information flow across bottlenecks. In this way, rewiring has a crisp complexity-theoretic meaning: it is the intervention that can move an instance across the boundary between "fast" and "slow" depth regimes.

We also emphasize that exact collision-free refinement is a stringent requirement that may be unnecessary in some downstream uses. Accordingly, we consider a bounded-error fingerprinting variant in which equal WL types must always agree, but unequal types are allowed to collide with small probability. This relaxation aligns with the standard practice of hashing large identifiers into fixed-width embeddings, while still maintaining a one-sided guarantee that preserves equality. In the RL-CONGEST setting, such fingerprints can often be produced with significantly shorter messages than exact global ranks, and they admit simple analyses via universal hashing once a canonical encoding of the local multiset is fixed. The resulting picture is that rewiring-to-expansion benefits both exact and approximate canonicalization, but the approximate regime can further reduce the communication footprint and is therefore the more plausible operational target when bandwidth is small.

Finally, we address the feasibility of rewiring itself. Selecting a small set of edges that optimally increases conductance is computationally intractable in general, as it subsumes hard cut-optimization problems. Nevertheless, for standard hard families that are archetypal obstructions to fast information spreading—paths, grids, and clustered graphs with sparse interconnections—we can characterize the minimal edge-addition budgets (up to polylogarithmic factors) necessary and sufficient to obtain constant conductance. These results clarify when rewiring can plausibly act as an accelerator under realistic constraints such as bounded per-node additions, and they motivate a computability-driven rewiring objective: rather than optimizing a task-specific surrogate, we target structural properties (notably expansion) that are directly tied to the round complexity of canonicalization primitives.

In summary, we contribute (i) an explicit reduction from one-step WL refinement to distributed sorting/ranking on the rewired graph, yielding polylogarithmic-depth algorithms under high conductance; (ii) a one-sided-error fingerprinting alternative with small sketches and comparable depth; (iii) conductance-parameterized lower bounds showing that collision-free refinement provably requires many rounds on low-conductance families; and (iv) rewiring budget characterizations for raising conductance on canonical hard graphs, supporting a phase-transition view of depth as a function of expansion. The subsequent sections formalize the RL-CONGEST-with-preprocessing model, relate conductance and mixing to routing/sorting guarantees on expanders, and position WL refinement as a canonicalization primitive that exposes these tradeoffs.

2 Background

We briefly recall the distributed model we work in, the semantics of one step of color refinement, and the expansion parameters that govern the complexity of global coordination primitives. Our intent is not to survey the full literature, but to isolate a small set of notions that will be used repeatedly: (i) the RL-CONGEST communication constraints, (ii) WL-1 refinement as a *canonicalization* task over locally formed keys, and (iii) conductance/mixing as the structural quantity controlling fast routing, sorting, and hence ranking.

RL-CONGEST and preprocessing. In the standard CONGEST model, computation proceeds in synchronous rounds on an undirected communication graph, where in each round each endpoint of an edge may send an $O(\log n)$-bit message across that edge. We work with a width-parameterized variant in which each edge transmits $O(w \log n)$ bits per round for a user-specified width $w \geq 1$; this captures the common situation in message-passing architectures where a larger hidden dimension permits proportionally larger per-edge communication. We further allow randomization (private coins) and local computation time $\mathrm{TIME}(\mathrm{poly}(\Delta', \log n))$ per round, where Δ' denotes the maximum degree of the actual communication graph used during message passing.

A central aspect of our setting is an explicit *preprocessing* stage that can modify the communication topology by adding edges. Concretely, a preprocessor receives (G, x) and outputs a rewired graph $G' = (V, E \cup E_+)$ subject to a budget constraint, either a global constraint $|E_+| \leq R$ or a per-node constraint $\deg_{E_+}(u) \leq r$ for all $u \in V$. After preprocessing, the message-passing phase runs for d rounds on G'. This separation lets us treat added edges as a resource distinct from depth, and it isolates the algorithmic question we care about: given that the post-rewiring bandwidth per edge is fixed (up to w), what properties of G' determine whether a canonicalization primitive can be completed in few rounds?

WL-1 refinement as a canonicalization problem. One iteration of the 1-dimensional Weisfeiler–Leman procedure (color refinement) transforms an initial coloring $x \in [p(n)]^n$ into a new coloring y by mapping each node u to a label that depends on its own color and the multiset of neighbor colors. We write

$$\mathrm{type}_G(u) \;:=\; \Big(x_u,\; \{\{x_v :\; v \in N_G(u)\}\}\Big).$$

The key point for us is that WL refinement is local in the formation of $\mathrm{type}_G(u)$ but global in the assignment of new labels. Under *perfect hashing semantics* (the usual mathematical definition), the output y must satisfy $y_u = y_v$ if and only if $\mathrm{type}_G(u) = \mathrm{type}_G(v)$. This is not merely an injective encoding requirement: it is a *global consistency* condition, because it requires that identical types computed at distant nodes be mapped to identical output identifiers, and that distinct types be separated everywhere.

In distributed terms, after one neighbor-exchange round each node can locally form a *key* k_u representing its type (e.g., by sorting the received neighbor colors and concatenating). The remaining task is to compute a globally consistent identifier for each distinct key value across the network. We view this as a canonicalization primitive: given distributed tokens $\{(k_u, u)\}_{u \in V}$, assign each node a label that depends only on k_u and is consistent across all nodes with the same key. A convenient instantiation is *ranking* (or distributed sorting): impose a total order on keys and return to each node the rank of its key among the distinct key values. Exact ranking yields collision-free WL refinement immediately. This viewpoint is operationally useful because ranking is a standard global primitive whose round complexity on a given graph can be analyzed via known routing and sorting subroutines.

We also distinguish an approximate alternative that will recur throughout: *fingerprints* (short sketches) that preserve equality deterministically but may allow collisions among unequal keys with small probability. Formally, each node outputs $h_u \in \{0,1\}^b$ such that $\mathrm{type}_G(u) = \mathrm{type}_G(v)$ implies $h_u = h_v$ always, while $\mathrm{type}_G(u) \neq \mathrm{type}_G(v)$ implies $\Pr[h_u = h_v] \leq \varepsilon$. In this regime the canonicalization target is weaker than exact ranking, but it remains one-sided and thus still captures the idea that equality of WL types must be globally coherent.

Conductance and mixing as coordination parameters. To relate the topology of G' to the complexity of canonicalization, we require a quantitative measure of how quickly information can spread. We use *conductance* $\phi(G')$, defined by

$$\phi(G') = \min_{S \subset V,\ 0 < \mathrm{vol}(S) \leq \mathrm{vol}(V)/2} \frac{|\partial S|}{\mathrm{vol}(S)},$$

where ∂S is the edge boundary and $\mathrm{vol}(S) = \sum_{u \in S} \deg_{G'}(u)$. Informally, large conductance rules out sparse cuts: every moderately sized set has many edges leaving it relative to its volume. This parameter is tightly connected to the mixing time $\tau_{\mathrm{mix}}(G')$ of the lazy random walk. In bounded-degree graphs, one may view τ_{mix} and $1/\phi$ as interchangeable up to polynomial factors; for example, standard Cheeger-type inequalities bound the spectral gap in terms of ϕ^2, and mixing time is $O(\log n/\mathrm{gap})$, yielding $\tau_{\mathrm{mix}}(G') \leq \widetilde{O}(1/\phi(G')^2)$ under mild regularity conditions. We will not rely on a specific inequality, but rather on the general principle that high conductance (or fast mixing) enables rapid global coordination, while low conductance imposes cut-capacity bottlenecks that no distributed protocol can bypass.

Routing, sorting, and ranking on expanders. The canonicalization view reduces WL refinement to a global ordering task over n keys of size $\Theta(\Delta' \log n)$ bits. In bandwidth-limited models, the classical route to such an ordering is to first obtain an efficient *routing* primitive on the communication graph, and then build sorting/ranking from routing. On graphs with constant conductance (often treated as expanders for algorithmic purposes), there exist randomized routing schemes with polylogarithmic round complexity that can deliver $O(1)$-congestion, $O(\text{polylog}(n))$-dilation routes for a large set of source–destination pairs. Typical constructions proceed by variants of Valiant-style load balancing: each token is sent to a (pseudo)random intermediate node and then to its destination, with random walks or expander properties ensuring near-uniform load and bounded congestion with high probability.

Once we can route tokens efficiently, we can implement distributed sorting by standard reductions: for instance, one may sample splitters, route tokens to buckets determined by splitters, recursively sort within buckets, and finally perform a global merge; alternatively, one can implement comparison-based sorting via a sorting network whose compare-exchange operations are realized by routing endpoints together. The details of the chosen primitive are not essential for the conceptual reduction, but two quantitative points are essential in our setting. First, the round complexity scales with the *routing time* $T_{\text{route}}(n, \phi)$ guaranteed by the graph, which is polylogarithmic in n when $\phi = \Omega(1)$ under standard randomized expander routing analyses. Second, the complexity must account for *token size*: since each WL key contains a multiset of up to Δ' colors, any exact token representation costs $\Theta(\Delta' \log n)$ bits, and thus each token must be streamed in $\Theta(\Delta'/w)$ batches when the per-edge bandwidth is $O(w \log n)$ bits. This yields a multiplicative factor Δ'/w in the number of rounds for sorting/ranking based on moving explicit keys (or, in some variants, on moving compact hash values together with collision-resolution logic).

Virtual edges and overlays. The preceding discussion suggests a conceptual interpretation of rewiring: by adding a relatively small number of edges, we aim to construct an *overlay* communication structure on top of the original graph that supports expander-like routing guarantees. There is a substantial algorithmic literature on simulating richer communication patterns via "virtual edges" or overlays, including approaches based on expander embeddings, low-congestion shortcuts, and sparse spanners/hopsets. While our model makes rewiring explicit by adding actual edges E_+, the role of these results is to justify the following heuristic: if we can endow G' with a sufficiently expanding overlay (either literally, by added edges, or effectively, by a structure that behaves like one for routing), then global primitives such as broadcast, aggregation, and sorting can be executed in polylogarithmic depth. Conversely, if no such overlay exists within the allowed budget, then there will be sparse cuts that limit the flow of information and force large depth for collision-free canonicalization tasks.

In particular, one should distinguish two effects. Adding edges may (a) increase $\phi(G')$ directly, improving the intrinsic routing properties of the communication graph, and/or (b) increase Δ' and hence the token-size and congestion costs of subsequent protocols. Our reductions make this tradeoff explicit through the dependence on Δ' and $\phi(G')$. This prepares the ground for the budgeted rewiring questions addressed later: we will quantify when a budget R or r suffices to move a hard family (e.g., a path or a clustered graph) into a constant-conductance regime, and we will analyze the resulting depth for implementing either exact WL refinement (via collision-free ranking) or the fingerprinted variant (via short one-sided sketches).

Summary of the background reduction. We will repeatedly exploit the same structural pipeline: one round of local exchange forms WL keys; the remainder of the computation is a global canonicalization of these keys. On high-conductance graphs, canonicalization can be realized via expander routing/sorting in polylogarithmic rounds, with an explicit Δ'/w factor accounting for token size under bandwidth constraints. On low-conductance graphs, sparse cuts limit the cut capacity and obstruct collision-free global agreement on labels. The next section formalizes the precise tasks (exact refinement versus fingerprinting), the two budget models for rewiring, and the evaluation metrics used to state the resulting tradeoffs.

3 Problem Setup

We now formalize the objects we are allowed to change (the communication topology), the two refinement targets we study (exact versus fingerprinted), and the performance metrics used to state tradeoffs. Throughout, the input instance is an undirected graph $G = (V, E)$ with $n = |V|$, $m = |E|$, maximum degree Δ, and an initial coloring $x \in [p(n)]^n$. The message-passing phase is an RL-CONGEST protocol with width parameter w, meaning that in each round each endpoint of an edge may transmit an $O(w \log n)$-bit message across that edge.

Two rewiring-budget models. A preprocessing algorithm Preprocess receives (G, x) and outputs a rewired graph

$$G' = (V, E \cup E_+),$$

where E_+ is a set of added undirected edges. We will write Δ' for the maximum degree of G'. We consider two budget constraints.

(Global budget.) The preprocessor must satisfy

$$|E_+| \;\le\; R.$$

This model captures a setting in which we are allowed to add a fixed number of auxiliary communication links (or long-range attention edges), with no explicit fairness constraint across nodes.

(Per-node budget.) The preprocessor must satisfy

$$\deg_{E_+}(u) \;\le\; r \qquad \text{for all } u \in V.$$

This captures degree-capped rewiring, as in bounded fan-out overlay construction or architectures where each node may attend to only r additional nodes. In this model, Δ' is automatically bounded by $\Delta + r$, but the global number of added edges may be as large as $nr/2$.

In either budget model, we allow Preprocess to be randomized and (unless otherwise stated) to be centralized in the sense that it may inspect the entire input graph. Our upper and lower bounds are stated *conditioned on* the resulting structural parameters of G', most prominently its conductance $\phi(G')$, and thus separate the graph-theoretic question "what conductance can we achieve under a given budget?" from the distributed-algorithmic question "what depth suffices given that conductance?". The preprocessing stage may also attach auxiliary node features of size $\mathrm{poly}(\Delta', \log n)$, which we treat as free local advice once installed; theorems will not rely on such advice except for disseminating (or fixing) short public randomness when convenient.

What is being refined: semantics versus communication. The *semantic* neighborhood used to define WL types is the original input graph G. That is, $\mathrm{type}_G(u) = (x_u, \{\{x_v : v \in N_G(u)\}\})$ is defined exactly as in Section **??**. Rewiring does not change the target partition of V into types; it only augments the communication graph on which we compute. This choice isolates the role of E_+ as a coordination resource rather than as a modification of the combinatorial object whose WL refinement we intend to compute. (In applications one may instead define types with respect to G'; our analysis applies verbatim after replacing G by the chosen refinement graph, provided the requisite one-round neighborhood exchange is performed on that graph.)

Because $E \subseteq E \cup E_+$, every original edge remains available for message passing. In particular, the one-round neighbor exchange required to locally form $\mathrm{type}_G(u)$ can always be executed over G' by restricting attention to the original adjacency. Subsequent rounds may freely exploit E_+ for global coordination.

Task I: exact WL refinement (collision-free canonicalization). In the exact task, after preprocessing and d rounds of RL-CONGEST on G', each node u must output an integer label y_u such that

$$y_u = y_v \iff \mathrm{type}_G(u) = \mathrm{type}_G(v). \tag{1}$$

We interpret (1) as "perfect hashing semantics": the labeling must be globally consistent (equal types map to equal labels everywhere) and collision-free (unequal types never share a label). The range of labels is unconstrained beyond being polynomially bounded; concretely, it suffices that $y_u \in [p''(n)]$ for some fixed polynomial p'', which is consistent with computing ranks among distinct keys.

Randomization is allowed, but the correctness requirement is *zero error*: for every fixed input (G, x) and every outcome of the preprocessor, the message-passing protocol must satisfy (1) with probability 1 over its private coins. When we later state randomized upper bounds for canonicalization primitives (e.g., routing/sorting on expanders), we will therefore be explicit about whether the resulting refinement is exact (probability 1) or succeeds with high probability; when needed, we will amplify success and then determinize collision resolution so that the final WL labeling remains collision-free.

Task II: fingerprinted WL refinement (one-sided bounded error). The second task relaxes the output to a short sketch while preserving one-sided correctness. Each node outputs a bitstring $h_u \in \{0,1\}^b$ such that

$$\mathrm{type}_G(u) = \mathrm{type}_G(v) \Longrightarrow h_u = h_v \qquad \text{always,} \tag{2}$$

$$\mathrm{type}_G(u) \neq \mathrm{type}_G(v) \Longrightarrow \Pr[h_u = h_v] \leq \varepsilon. \tag{3}$$

The probability in (3) is over the private randomness of the protocol (and any randomness used in preprocessing, if applicable). Condition (2) is deterministic and enforces global coherence of equality: identical WL types must *never* be separated by the sketching procedure. Condition (3) allows a small collision probability among distinct types. We will target sketch lengths $b = O(\log(n/\varepsilon))$, consistent with standard universal hashing guarantees once a canonical encoding of $\mathrm{type}_G(u)$ is fixed.

One may view fingerprinted WL as implementing a randomized map H from keys to short strings where equality is preserved by construction (e.g., by hashing a deterministic serialization of the key), and the only source of error is accidental collision of two unequal serializations. This task is strictly weaker than exact refinement, but it is the appropriate objective when we wish to isolate how conductance accelerates global agreement and when we are willing to tolerate a small probability of false equality.

Evaluation metrics and parameter dependence. We measure the complexity of an algorithm by the tuple

$$(d,\ w,\ R) \quad \text{or} \quad (d,\ w,\ r),$$

together with the achieved post-rewiring parameters $(\Delta', \phi(G'))$ and, in the fingerprinted task, the pair (b, ε). The primary metric is the *depth* d, the number of synchronous message-passing rounds on G'. Width w is treated as a fixed architectural parameter that linearly scales per-edge bandwidth, and the dependence of d on w is central: since explicit WL keys have $\Theta(\Delta' \log n)$ bits, many primitives incur a multiplicative streaming factor $\Theta(\Delta'/w)$.

We distinguish two notions of probability of failure. For exact refinement we require zero error, as discussed above. For fingerprinting, we parameterize failure by ε as in (3). When we state "with high probability" guarantees for intermediate routing/sorting subroutines, we will mean probability at least $1 - n^{-c}$ for an arbitrary fixed constant $c > 0$, and we will account for union bounds across all nodes and across all comparisons implicit in sorting/ranking. In particular, if a protocol has internal randomness but is intended to satisfy (2) deterministically, we will enforce that determinism at the level of the canonical encoding step and confine randomness to the final hashing.

Finally, since rewiring may increase degrees, we explicitly track Δ' as part of the evaluation: higher Δ' can facilitate expansion but also increases local key size, congestion, and memory requirements. Our upper bounds will therefore be stated as functions of $(w, \Delta', \phi(G'), n)$, while our budget theorems will quantify when a given (R or r) can force $\phi(G')$ into a desired regime without making Δ' prohibitively large.

Problem statement (compressed). Given (G, x) and parameters specifying either a global rewiring budget R or a per-node budget r, plus a width w and (for fingerprints) an error target ε, our goal is to design Preprocess and an RL-CONGEST protocol on the resulting G' that accomplishes either exact refinement (1) or fingerprinting (2)–(3) in as few rounds d as possible. The next sections analyze this goal in two complementary directions: upper bounds that reduce refinement to global ranking on high-conductance G', and lower bounds showing that sparse cuts (hence small $\phi(G')$) impose unavoidable depth for collision-free canonicalization.

4 Upper Bounds I: Exact WL via Sorting on High-Conductance Graphs

We now give an explicit upper bound for one iteration of $\mathrm{WL1}(G, x)$ under the exact (collision-free) semantics, assuming that preprocessing has produced a communication graph G' with sufficiently large conductance. The proof proceeds by a reduction from WL refinement to a global *ranking* problem on node-local keys, followed by an invocation of standard routing/sorting primitives on graphs with $\phi(G') \geq \phi$.

From WL refinement to global canonical ranks. Fix the semantic types with respect to G, namely

$$\mathrm{type}_G(u) \;=\; \big(x_u,\ \{\{x_v : v \in N_G(u)\}\}\big).$$

After a single neighbor exchange restricted to the original edges E (performed over G'), each node u can locally form a concrete representation of $\mathrm{type}_G(u)$. For definiteness, we assume that u sorts the received neighbor colors and stores the resulting list; this yields a canonical string encoding

$$\mathrm{enc}(u) \in \{0,1\}^{\Theta(\deg_G(u)\log n)} \subseteq \{0,1\}^{\Theta(\Delta'\log n)}.$$

The exact WL objective is to assign labels y_u so that $y_u = y_v$ iff $\mathrm{enc}(u) = \mathrm{enc}(v)$. This is precisely the task of computing a *collision-free canonical name* for each distinct string among $\{\mathrm{enc}(u)\}_{u\in V}$.

A convenient canonicalization is obtained by assigning the *rank* of $\mathrm{enc}(u)$ among all distinct encodings under lexicographic order. Concretely, if we let $\mathcal{K} = \{\mathrm{enc}(u) : u \in V\}$ denote the set of realized keys and $\pi : \mathcal{K} \to [|\mathcal{K}|]$ be the order-preserving bijection, then setting

$$y_u \;:=\; \pi(\mathrm{enc}(u))$$

solves the exact refinement task. Thus, it suffices to implement distributed ranking of node-held keys. Importantly, the range $[|\mathcal{K}|] \subseteq [n]$ is polynomially bounded, so labels fit in $O(\log n)$ bits once computed.

Tokenization and the sorting interface. We reduce ranking to distributed sorting of *tokens*. Each node u creates a token

$$t_u \;=\; (\mathrm{enc}(u), \mathrm{ID}(u)),$$

where $\mathrm{ID}(u)$ is a unique tie-breaker. We stress that tie-breaking is not part of the WL semantics; it is used only to induce a strict total order among tokens so that a distributed sorter can be viewed as returning a permutation of $\{t_u\}_{u\in V}$. Once tokens are sorted, we can recover ranks of *keys* (rather than of individual tokens) by an additional local comparison step: each token compares enc with its predecessor in the sorted order and increments a counter exactly when the key changes. Standard reductions implement this conversion from sorted order to distinct-key ranking with $O(1)$ additional routing passes.

The bottleneck is token size. Since $\mathrm{enc}(u)$ is $\Theta(\deg_G(u)\log n)$ bits, and we allow $\deg_G(u) \le \Delta'$, we must accommodate tokens of size

$$L \;=\; \Theta(\Delta'\log n)\ \text{ bits.}$$

Under width w, each edge transmits only $O(w\log n)$ bits per round, so a token must be *streamed* in

$$s \;=\; \Theta(\Delta'/w)$$

batches. We will account for this factor multiplicatively in the round complexity.

Routing/sorting on graphs with $\phi(G') \geq \phi$. We abstract the underlying distributed primitive as follows. Let $T_{\mathrm{route}}(n, \phi)$ denote the number of rounds required to route n unit-size tokens (i.e., $O(\log n)$ bits each) with $O(1)$ congestion and $O(1)$ dilation on an n-node graph of conductance at least ϕ, under RL-CONGEST bandwidth $O(\log n)$ per edge per round. Numerous constructions and analyses yield $T_{\mathrm{route}}(n, \phi) = \mathrm{polylog}(n)$ for $\phi = \Omega(1)$, typically via Valiant-style random intermediate destinations combined with rapid mixing; for ϕ bounded away from 1, one often obtains a dependence of the form $\mathrm{polylog}(n)/\phi^c$ for a constant $c > 0$ dictated by the routing scheme.

Given such routing, we obtain sorting by standard reductions (routing $\Rightarrow$ compare-exchange networks $\Rightarrow$ sorting $\Rightarrow$ ranking). We will treat this as a black box ExpanderSort, which takes as input one token per node and outputs (implicitly, via final token positions) the sorted order. The reduction is classical: if we can realize an appropriate sequence of matchings or random permutations in T_{route} rounds each, then we can simulate a sorting network of polylogarithmic depth, paying a polylogarithmic overhead in n and a polynomial overhead in $1/\phi$ if the routing time depends on ϕ.

Depth bound with streaming. Combining the streaming factor $s = \Theta(\Delta'/w)$ with the unit-token sorting cost yields the following bound.

Theorem 4.1 (Exact WL via sorting on high-conductance G')**.** *Assume that preprocessing outputs a communication graph G' with maximum degree Δ' and conductance $\phi(G') \geq \phi$. There is an RL-CONGEST protocol that computes one exact WL refinement step (collision-free labels satisfying* (1)*) in*

$$d \;\leq\; \widetilde{O}\left(\frac{\Delta'}{w} \cdot T_{\mathrm{route}}(n, \phi)\right)$$

rounds, with local computation and memory bounded by $\mathrm{poly}(\Delta', \log n)$.

We emphasize the provenance of each parameter. The factor Δ'/w is unavoidable for exact canonicalization when keys explicitly contain neighbor multisets: even if global coordination were free, node u must (in some form) move $\Theta(\deg_G(u) \log n)$ bits through edges of capacity $O(w \log n)$ to participate in a global comparison-based procedure. The remaining term $T_{\mathrm{route}}(n, \phi)$ captures purely *topological* limitations of G' as a communication medium.

Exactness and zero-error semantics. Theorem 4.1 is stated for the exact WL objective, which requires that no two distinct types ever collide. When ExpanderSort and its underlying routing are randomized and succeed only with high probability, we obtain a Monte Carlo protocol for ranking; this is insufficient on its face for zero-error refinement. To reconcile this, we use the following standard device: after the purported ranking is produced, we perform a verification pass that deterministically checks *consistency* of the output with the sorted order of keys. Concretely, the protocol routes adjacent pairs in the claimed order to a common endpoint (or routes succinct checksums along the same ordering structure) and verifies that (i) the order is nondecreasing in enc and (ii) equal keys induce equal ranks. Any detected violation triggers a restart with fresh private randomness (or, if one prefers worst-case termination, a fallback to a slower deterministic canonicalization routine). This yields a Las Vegas implementation with zero error and expected round complexity within a constant factor of the high-probability bound, since the failure probability can be made n^{-c} for an arbitrary constant c and the verification cost is subsumed by the sorting cost up to polylogarithmic factors.

Discussion: conductance as the coordination enabler. The salient point is that the WL task is locally easy and globally hard. Local type formation is completed in one round over E; all subsequent difficulty is in agreeing on a globally collision-free naming of possibly many distinct local keys. High conductance of G' precisely supplies the missing ingredient: it supports near-uniform dispersion of information (hence fast routing), which in turn supports fast sorting/ranking. In the next section we show that if we relax collision-freeness to one-sided bounded-error fingerprints, then we can bypass global perfect hashing and obtain polylogarithmic depth under weaker coordination requirements, while still preserving the guarantee that equal WL types are never separated.

5 Upper Bounds II: One-Sided-Error WL Fingerprints Without Global Perfect Hashing

We now relax the exact WL objective by allowing *fingerprints* of WL types with one-sided error. Formally, each node u must output a sketch $h_u \in \{0,1\}^b$ such that equal WL types are *never* separated,

$$\text{type}_G(u) = \text{type}_G(v) \implies h_u = h_v \quad \text{(always)},$$

while unequal types collide with probability at most ε,

$$\text{type}_G(u) \neq \text{type}_G(v) \implies \Pr[h_u = h_v] \leq \varepsilon,$$

where the probability is over the internal randomness of the protocol. The crucial structural difference from the exact task is that we no longer need a globally consistent collision-free naming of *all* realized keys; it suffices to agree on a *single shared hash function* and apply it locally. This bypasses the global sorting/ranking subroutine and the attendant perfect-hashing semantics.

Canonical local encoding. As in the exact case, a single neighbor exchange over the original edges E suffices for each node to learn the multiset of neighbor colors in G. We again let $\mathrm{enc}(u)$ denote a canonical string encoding of $\mathrm{type}_G(u)$ obtained by sorting the received neighbor colors and concatenating them with x_u (and, if desired for robustness, with $\deg_G(u)$). We only require the following deterministic property:

$$\mathrm{type}_G(u) = \mathrm{type}_G(v) \quad \Longleftrightarrow \quad \mathrm{enc}(u) = \mathrm{enc}(v).$$

This local canonicalization uses only $\mathrm{poly}(\Delta', \log n)$ time and memory (sorting $\deg_G(u) \le \Delta'$ items of $O(\log n)$ bits each). No further communication is needed to construct $\mathrm{enc}(u)$.

One-sided fingerprints via shared hashing. Let $\mathcal{H}$ be a family of hash functions mapping binary strings of length at most $\Theta(\Delta' \log n)$ to $\{0,1\}^b$. A shared random seed σ selects $H_\sigma \in \mathcal{H}$, and each node outputs

$$h_u \;:=\; H_\sigma(\mathrm{enc}(u)).$$

Because H_σ is a deterministic function given σ, the implication $\mathrm{enc}(u) = \mathrm{enc}(v) \Rightarrow h_u = h_v$ holds *unconditionally*, yielding the required one-sided guarantee. The only failure event is a collision on two distinct encodings, whose probability is controlled by the universality of $\mathcal{H}$.

A convenient instantiation is a Karp–Rabin style fingerprint for variable-length strings. Fix a prime q and interpret $\mathrm{enc}(u)$ as a sequence of $O(\Delta')$ integers in $[p(n)]$ (or $O(\Delta' \log n)$ bits). Choose a uniform $\alpha \in \mathbb{F}_q$ and define

$$H_\alpha(\mathrm{enc}) \;:=\; \sum_{i=0}^{\ell-1} \mathrm{enc}[i] \cdot \alpha^i \bmod q,$$

where ℓ is the length of the sequence. For distinct sequences, the collision probability satisfies $\Pr[H_\alpha(a) = H_\alpha(b)] \le \ell/q$ by the standard polynomial identity bound. Choosing $q \ge \Theta(\Delta'/\varepsilon) \cdot \mathrm{poly}(n)$ makes this probability at most ε, and the resulting sketch length is

$$b \;=\; \Theta(\log q) \;=\; O(\log(n/\varepsilon))$$

since $\Delta' \le \mathrm{poly}(n)$ in our regime (and any additional $\log \Delta'$ factor is absorbed into $\widetilde{O}(\cdot)$ notation). The hash can be computed in one pass over the locally stored encoding in $\mathrm{poly}(\Delta', \log n)$ time.

Coordinating the seed. The remaining question is how nodes obtain the shared seed σ (e.g., the choice of q and α). We allow two standard options consistent with our preprocessing model.

(i) Seed as a preprocessing feature. If the preprocessor is centralized (or otherwise permitted to attach bounded-size auxiliary features), it can simply write the same σ into every node. This costs no communication rounds.

(ii) Seed dissemination on G'. Otherwise, we disseminate σ after rewiring using standard rumor-spreading/broadcast mechanisms on graphs of conductance at least ϕ. Concretely, we may elect an arbitrary initiator (e.g., the minimum ID via any leader election available in the model, or a designated node given in the input variant) which samples σ and broadcasts it. Let $T_{\mathrm{bcast}}(n, \phi)$ denote the required rounds to broadcast an $O(\log(n/\varepsilon))$-bit message on G' with failure probability n^{-c}. On constant-conductance graphs one obtains $T_{\mathrm{bcast}}(n, \phi) = \mathrm{polylog}(n)$ by standard push–pull gossip analyses, and more generally $T_{\mathrm{bcast}}(n, \phi) = \widetilde{O}(1/\phi^{c'})$ for a scheme-dependent constant $c' > 0$. The bandwidth w enters only through the time to pipeline the $O(\log(n/\varepsilon))$ bits of σ across edges of capacity $O(w \log n)$ bits/round; this contributes an additional factor of $\left\lceil \frac{\log(n/\varepsilon)}{w \log n} \right\rceil$, which is $O(1)$ for constant w and polynomially small ε.

Round complexity and the polylogarithmic regime. Putting the pieces together, the depth is the sum of (a) one round to exchange colors along E (feasible on G' since $E \subseteq E(G')$), (b) $T_{\mathrm{bcast}}(n, \phi)$ rounds to coordinate σ if needed, and (c) zero additional rounds for hashing. In particular, on $\phi(G') = \Omega(1)$ we obtain polylogarithmic depth without performing any global sorting of keys.

Theorem 5.1 (One-sided WL fingerprints on high-conductance graphs). *Assume preprocessing outputs G' with conductance $\phi(G') \geq \phi$ and maximum degree Δ'. For every $\varepsilon \in (0, 1)$ there exists a randomized RL-CONGEST protocol that outputs sketches $h_u \in \{0, 1\}^b$ with*

$$b \;=\; O(\log(n/\varepsilon)),$$

such that $\mathrm{type}_G(u) = \mathrm{type}_G(v) \Rightarrow h_u = h_v$ *always and* $\mathrm{type}_G(u) \neq \mathrm{type}_G(v) \Rightarrow \Pr[h_u = h_v] \leq \varepsilon$*. The protocol runs in*

$$d \;\leq\; 1 \;+\; \widetilde{O}\Big(T_{\mathrm{bcast}}(n, \phi) \cdot \left\lceil \frac{\log(n/\varepsilon)}{w \log n} \right\rceil\Big)$$

rounds and uses local computation/memory $\mathrm{poly}(\Delta', \log n)$*. In particular, if $\phi = \Omega(1)$ and $w = \Theta(1)$, then $d = \widetilde{O}(\mathrm{polylog}(n))$.*

Tradeoffs: bits versus rounds versus collision probability. Theorem 5.1 captures the cleanest regime: we pay polylogarithmic time to coordinate a single seed and then obtain fingerprints in one local computation. More generally, we can tune ε against the number of communicated bits in the seed.

First, for any 2-universal family with collision probability exactly 2^{-b} (for distinct inputs), it suffices to set $b = \lceil \log(1/\varepsilon) \rceil$ to achieve $\Pr[h_u = h_v] \le \varepsilon$ for any fixed pair (u, v). If instead we wish to guarantee that *no* collision occurs among all unequal-type pairs with probability at least $1 - \varepsilon$, then a union bound over at most n^2 pairs yields the sufficient choice $b = \Theta(\log(n/\varepsilon))$, matching the stated parameterization.

Second, when w is small, it is sometimes preferable to send several short independent seeds over multiple rounds rather than one longer seed. If we compute k independent b_0-bit hashes and concatenate them, then the collision probability drops to at most 2^{-kb_0} while the total seed length scales linearly with k. Under width w, the dissemination time scales with $\left\lceil \frac{kb_0}{w \log n} \right\rceil \cdot T_{\text{bcast}}(n, \phi)$, exhibiting an explicit bits-to-rounds tradeoff. In the constant-conductance regime, this yields a smooth interpolation between (i) constant-round seed spread with moderate ε and (ii) polylog-round seed spread with negligible ε.

When bounded error is algorithmically decisive. The essential gain of fingerprinting is that we replace *global canonical naming* by *shared randomness plus local hashing*. Consequently, on well-connected G' the depth is governed by dissemination of $O(\log(n/\varepsilon))$ bits rather than by routing the full $\Theta(\Delta' \log n)$-bit keys through the network for global comparison. This is precisely the regime in which bounded error enables polylogarithmic depth even when exact refinement would be dominated by global coordination costs. In the next section we show that, in contrast, if we insist on collision-free refinement, then conductance constraints impose information-flow barriers of the form $\Omega((1/\phi) \cdot m/(w \log n))$, establishing a sharp separation between exact canonicalization and one-sided fingerprinting.

6 Lower Bounds Parameterized by Conductance

We next justify the dependence on expansion parameters in our upper bounds by proving that sparse cuts impose unavoidable information-flow constraints for *collision-free* refinement. Conceptually, the obstruction is not the local formation of WL keys—each node can form $\text{type}_G(u)$ after one neighbor exchange—but rather the requirement of a *globally consistent* collision-free naming (perfect-hashing semantics) of all realized keys. This is a canonicalization problem: it forces the network to reconcile which keys coincide across distant regions and to assign distinct names to all remaining keys.

A cut-capacity simulation lemma. Let $S \subset V$ and write $\bar{S} = V \setminus S$. We consider the standard two-party simulation in which Alice controls all nodes in S and Bob controls all nodes in $\bar{S}$, and the parties simulate an RL-CONGEST protocol round by round. Messages transmitted on edges with both endpoints in S (resp. in $\bar{S}$) are local to Alice (resp. Bob) and incur no two-party communication; only messages on the cut edges ∂S require communication between the parties.

Lemma 6.1 (Cut simulation)**.** *Let* Π *be any (possibly randomized) RL-CONGEST protocol running for* d *rounds on a graph* $G' = (V, E')$ *with per-edge bandwidth* $O(w \log n)$ *bits/round. Then for every* $S \subset V$ *there is a public-coin two-party protocol that simulates* Π *with total communication*

$$O\big(d \cdot |\partial S| \cdot w \log n\big)$$

bits, and reproduces the entire joint distribution of the transcript and all node outputs.

The lemma is immediate from the model definition: in each round and for each cut edge, at most $O(w \log n)$ bits cross between S and $\bar{S}$. In particular, for any task whose two-party communication complexity (under an appropriate input distribution) is $\Omega(B)$ bits, Lemma 6.1 yields the round lower bound

$$d \;=\; \Omega\!\left(\frac{B}{|\partial S| \cdot w \log n}\right). \tag{4}$$

To relate $|\partial S|$ to conductance, we recall that for the minimizer $S^\star$ in the definition of $\phi(G')$ we have $|\partial S^\star| = \phi(G') \cdot \mathrm{vol}(S^\star)$ and $\mathrm{vol}(S^\star) \leq \mathrm{vol}(V)/2$. On bounded-degree families one may equivalently regard $\mathrm{vol}(S) = \Theta(|S|)$, but we retain the volume formulation since it is the appropriate notion for general Δ'.

A canonicalization primitive subsumed by collision-free WL. We isolate a minimal primitive capturing the global coordination inherent in perfect hashing. Each node u holds a *key* k_u (in our application, a canonical encoding of $\mathrm{type}_G(u)$), and the network must output names $y_u \in [p''(n)]$ satisfying

$$y_u = y_v \iff k_u = k_v,$$

with *zero* error. The crucial point is that the name space is only $\mathrm{poly}(n)$, i.e., each y_u has $O(\log n)$ bits, whereas k_u may have $\Theta(\Delta' \log n)$ bits. Any solution must therefore implement an implicit *collision-free compression* of the realized keys, which in turn forces the protocol to discover enough global structure to avoid collisions among distinct keys that may be separated by a sparse cut.

Collision-free WL refinement with perfect hashing semantics is at least as hard as this primitive: given that each node can locally compute a canonical $k_u = \mathrm{enc}(u)$ for its WL type, producing the WL color y_u is precisely producing collision-free names for these keys.

Reduction from equality across a conductance-controlled cut. We now choose a graph family and an input distribution on initial colors x such that any collision-free naming of WL keys solves an instance of the two-party equality problem. Fix a parameter $\phi \in (0, 1)$ and a width $w = o(n/\log n)$. We construct $G_{n,\phi}$ by taking two bounded-degree expanders A and B on $\Theta(n)$ nodes each, and connecting them by a sparse cut with

$$|\partial A| \;=\; \Theta(\phi n),$$

so that $\phi(G_{n,\phi}) = \Theta(\phi)$. (One may realize this, for example, by adding $\Theta(\phi n)$ random edges between A and B and using standard expansion estimates for the resulting two-cluster graph; the precise construction is not essential for the reduction.)

We encode Alice's and Bob's inputs as follows. Let $L = \Theta(m)$, where $m = |E(G_{n,\phi})| = \Theta(n)$ for bounded degree. Inside each side, we place L designated "record" nodes whose WL keys will represent the input bits. Concretely, for each $i \in [L]$ we create a constant-size local gadget attached to a record node $a_i \in A$ (resp. $b_i \in B$) so that, after the one-round neighbor exchange needed to form $\mathrm{type}_G(\cdot)$, the resulting WL type of a_i encodes the ith input symbol of Alice, and similarly for b_i and Bob. These gadgets use only constant degree and can be made pairwise vertex-disjoint, so they do not affect conductance beyond constant factors.

The key property is that we arrange the encoding so that under the equality distribution,

$$\text{if } \alpha = \beta \in \{0,1\}^L, \quad \text{then } \{\mathrm{type}_G(a_i) : i \in [L]\} = \{\mathrm{type}_G(b_i) : i \in [L]\}$$

as multisets, whereas if $\alpha \neq \beta$ then (with probability 1 under the distribution) the two multisets differ on $\Omega(L)$ positions and hence contain $\Omega(L)$ keys that appear on exactly one side. Thus, a collision-free global naming of all WL types implicitly determines whether the two key multisets coincide.

To convert this into a two-party decision, we designate a node $z \in A$ whose WL key is chosen so that its canonical perfect-hash name depends on the global set of realized keys: if the multisets match, z must share its name with a corresponding node in B; if they do not match, z's name must be distinct from every name produced in B. This can be enforced by introducing a small number of additional "marker" keys that fix the relative position of z's key among all realized keys under the perfect-hashing semantics (e.g., via a globally agreed total order on encodings). Consequently, from the output label of z together with labels on a constant-size set of local markers in A, Alice can determine whether $\alpha = \beta$. We thus obtain that any zero-error collision-free WL refiner induces a two-party protocol for EQ_L.

Since randomized two-party equality has communication complexity $\Omega(L)$ even with public coins and constant error (and a fortiori for zero error), Lemma 6.1 and (4) imply

$$d = \Omega\left(\frac{L}{|\partial A| \cdot w \log n}\right).$$

With $L = \Theta(m)$ and $|\partial A| = \Theta(\phi n) = \Theta(\phi m)$ on our bounded-degree family, this yields the conductance-parameterized barrier claimed in Theorem 6.2 below (where we state the bound in the standard $\Omega((1/\phi) \cdot m/(w \log n))$ form, absorbing constant-factor relationships between m and n for this family).

Theorem 6.2 (Conductance lower bound for collision-free refinement)**.** *For every $\phi \in (0, 1)$ and every $w = o(n/\log n)$, there exists a family of graphs $\{G_{n,\phi}\}$ with maximum degree $\Delta = O(1)$ and conductance $\phi(G_{n,\phi}) = \Theta(\phi)$, together with a distribution over initial colorings $x \in [p(n)]^n$, such that any (possibly randomized) RL-CONGEST protocol that performs collision-free WL refinement with perfect-hashing semantics requires*

$$d = \Omega\left(\frac{1}{\phi} \cdot \frac{m}{w \log n}\right)$$

rounds on $(G_{n,\phi}, x)$.

Extensions to simpler canonicalization primitives. The same proof template applies to tasks strictly weaker than collision-free WL refinement, provided they still require nontrivial reconciliation of key sets across a sparse cut. Two examples are: (i) *global distinctness testing* (deciding whether some key appears on both sides of the cut), and (ii) *collision-free dictionary construction* (assigning distinct $O(\log n)$-bit IDs to distinct keys without requiring any further structure). In both cases, we may embed equality (or set disjointness) so that a correct output would decide the underlying communication problem, and the round complexity must again scale inversely with $|\partial S|$ and hence with $\phi(G')$.

Interpretation and contrast with fingerprints. Theorem 6.2 should be read as an information-flow obstruction to *zero-error* global canonicalization on graphs with sparse cuts: if $\phi(G')$ is small, then the cut capacity $|\partial S| \cdot w \log n$ is small, and any procedure that must rule out *all* collisions among distinct WL types is forced into large depth. This is precisely the point at which the one-sided-error fingerprinting objective becomes algorithmically decisive: once collisions are permitted with probability ε (while preserving the one-sided guarantee for equal types), the global canonicalization bottleneck disappears, and the depth can be driven instead by dissemination of an $O(\log(n/\varepsilon))$-bit seed, as in Section 5. In the next section we turn to the complementary question of how much rewiring is necessary and sufficient to raise $\phi(G')$ into the regime where such shallow protocols become possible.

7 Rewiring Budget vs. Achievable Conductance

We now quantify, on canonical low-expansion families, how many edges must be added in preprocessing in order to raise conductance into the regime where shallow refinement becomes possible. Throughout we view rewiring as adding a set E_+ of new edges, producing $G' = (V, E \cup E_+)$. We consider both a *global* budget $|E_+| \le R$ and a *per-node* budget $\deg_{E_+}(u) \le r$ for all u.

A general bottleneck accounting. Fix any set $S \subset V$ with $0 < \mathrm{vol}_{G'}(S) \le \mathrm{vol}_{G'}(V)/2$. Since only edges with exactly one endpoint in S contribute to $\partial_{G'} S$, we have

$$|\partial_{G'} S| \;=\; |\partial_G S| \;+\; |\{\, e \in E_+ : |e \cap S| = 1 \,\}|. \tag{5}$$

In particular, for any fixed S, the boundary can increase by at most R, while the volume $\mathrm{vol}_{G'}(S) = \mathrm{vol}_G(S) + 2|E_+[S]| + |\{e \in E_+ : |e \cap S| = 1\}|$ is nondecreasing and may increase even if added edges do *not* cross the cut. Consequently, for lower bounds it is conservative to pretend that *all* added edges cross the target bottleneck (since internal edges only make $\phi(G')$ smaller for that S).

The standard way we will use (5) is: identify a natural bisection S for which $|\partial_G S|$ is small but $\mathrm{vol}_G(S)$ is large (typically $\Theta(n)$ on bounded-degree graphs). Then any rewiring with $R = o(\mathrm{vol}_G(S))$ leaves

$$\frac{|\partial_{G'} S|}{\mathrm{vol}_{G'}(S)} \;\le\; \frac{|\partial_G S| + R}{\mathrm{vol}_G(S)} \;=\; o(1),$$

and hence cannot yield $\phi(G') = \Omega(1)$.

Paths. Let $G = P_n$ be the path on n nodes. Take S to be the first $\lfloor n/2 \rfloor$ nodes along the path. Then $|\partial_G S| = 1$, while $\mathrm{vol}_G(S) = \Theta(n)$ (indeed $\mathrm{vol}_G(S) = 2|S| - 1$). For any rewiring budget R we obtain

$$\phi(G') \;\le\; \frac{|\partial_{G'} S|}{\mathrm{vol}_{G'}(S)} \;\le\; \frac{1 + R}{\Theta(n)}. \tag{6}$$

Therefore, achieving $\phi(G') = \Omega(1)$ on P_n requires $R = \Omega(n)$. Moreover, (6) shows a sharp impossibility when $R = o(n)$: even if we place added edges adversarially and all of them cross the midpoint cut, the conductance remains $o(1)$.

This lower bound is essentially tight. One can add an explicit constant-degree expander overlay $H = (V, F)$ on the same vertex set and set $E_+ = F$, giving $R = \Theta(n)$ and $\deg_{E_+}(u) = O(1)$ for all u. On bounded-degree bases such as P_n, the union $G' = G \cup H$ then has $\phi(G') = \Omega(1)$: for every S with $|S| \le n/2$ we have $|\partial_{G'} S| \ge |\partial_H S| = \Omega(|S|)$ and $\mathrm{vol}_{G'}(S) = O(|S|)$, hence $|\partial_{G'} S| / \mathrm{vol}_{G'}(S) = \Omega(1)$.

Grids. Let G be the $\sqrt{n} \times \sqrt{n}$ grid (assume n is a perfect square for simplicity). Let S be the left half of the grid (the first $\sqrt{n}/2$ columns). Then $|\partial_G S| = \Theta(\sqrt{n})$ (the vertical cut) while $\mathrm{vol}_G(S) = \Theta(n)$ since the grid has bounded degree. Thus any rewiring with $|E_+| \le R$ satisfies

$$\phi(G') \;\le\; \frac{\Theta(\sqrt{n}) + R}{\Theta(n)}. \tag{7}$$

In particular, to obtain $\phi(G') = \Omega(1)$ we must have $R = \Omega(n)$. As in the path case, $R = o(n)$ forces $\phi(G') = o(1)$, independent of how edges are chosen.

Again the bound is near-tight: adding a bounded-degree expander overlay uses $R = \Theta(n)$ and yields constant conductance while keeping $\Delta' = \Delta + O(1)$. Alternatively, adding a sparse random overlay $G(n, p)$ with $p = \Theta(\log n / n)$ uses $R = \Theta(n \log n)$ and yields $\phi(G') = \Omega(1)$ with high probability, at the cost of increasing Δ' by $\Theta(\log n)$ with high probability. The expander-overlay option is preferable if we treat Δ' as a first-class resource, since our refinement depth scales linearly with Δ'/w in the routing-based upper bounds.

Two-community graphs. Consider the archetypal “two-cluster” obstruction: let A and B be two vertex-disjoint constant-degree expanders on $n/2$ nodes each, and connect them by a single edge. Let $S = A$. Then $|\partial_G S| = 1$ while $\mathrm{vol}_G(S) = \Theta(n)$. By the same accounting as for paths, any rewiring with budget R satisfies

$$\phi(G') \;\le\; \frac{1 + R}{\Theta(n)}. \tag{8}$$

Thus $R = \Omega(n)$ is necessary to raise conductance to a constant. Intuitively, constant expansion requires $\Theta(n)$ edges crossing every balanced cut, and the original graph contributes only $O(1)$ such edges across the planted partition.

Sufficiency with $R = \Theta(n)$ again follows by adding a constant-degree expander overlay on all n nodes, which in particular inserts $\Omega(n)$ crossing edges between A and B in a pseudorandomly dispersed way, eliminating the sparse inter-cluster bottleneck.

Global vs. per-node budgets and degree blow-up. The preceding sufficiency statements highlight a key design choice: we may spend the budget either to create a *low-degree* expander overlay (keeping Δ' bounded, hence keeping the factor Δ'/w small), or to add *random* edges independently, which typically requires larger average added degree to achieve comparable guarantees without coordination.

To make this contrast explicit, consider the two-cluster family above and suppose rewiring is *oblivious* in the following sense: each node selects its r new neighbors independently and uniformly from V (subject to avoiding duplicates), so $R = \Theta(nr)$. Let X be the number of added edges that cross (A, B). Under this model, $\mathbb{E}[X] = \Theta(R/2) = \Theta(nr)$, but concentration at the scale needed for conductance is more stringent because we require $X = \Omega(n)$ *and* we require that no other set S forms a competing bottleneck. A standard union bound over cuts together with Chernoff bounds yields that achieving $\phi(G') = \Omega(1)$ with high probability via independent random choices typically demands $r = \Theta(\log n)$ (equivalently $R = \Theta(n \log n)$), matching the well-known threshold for random graphs to become expanders. In contrast, an explicit constant-degree expander overlay achieves the same conductance target with $r = O(1)$ and $R = \Theta(n)$, but it requires coordinated edge placement in preprocessing.

From the perspective of our subsequent refinement algorithms, the difference matters quantitatively: random overlays often incur $\Delta' = \Theta(\log n)$ whp, increasing the depth bound by a $\Theta(\log n)$ factor, whereas bounded-degree overlays preserve $\Delta' = \Theta(1)$ on bounded-degree inputs.

Summary: a near-tight phase boundary at linear budget. On the path, grid, and two-community families, the inequalities (6), (7), and (8) show that any sublinear global budget $R = o(n)$ leaves $\phi(G') = o(1)$, and hence cannot place the instance into the constant-conductance regime. Conversely, a linear budget $R = \Theta(n)$ is sufficient (up to constant factors) to force $\phi(G') = \Omega(1)$ via bounded-degree expander overlays, thereby enabling the polylogarithmic-depth refinement procedures described earlier when w is not too small. This establishes a robust "linear-budget barrier" for repairing expansion on these canonical hard families: below $\Theta(n)$ added edges, constant conductance is information-theoretically unattainable; at $\widetilde{O}(n)$ edges, it becomes achievable with careful construction.

In the next section we turn from worst-case feasibility to *computability-driven* objectives: since optimal conductance improvement under a strict budget is generally intractable, we propose efficiently estimable proxies that correlate with routability and thus with the achievable refinement depth.

8 Computability-Driven Rewiring Objectives

We now formalize the algorithmic tension implicit in the preceding section: while our refinement procedures benefit from large $\phi(G')$ (equivalently fast mixing and good routability), selecting a set of at most R added edges that maximizes $\phi(G')$ is intractable in general (cf. the hardness discussion in the global context). Hence, rather than optimizing conductance directly, we propose *computability-driven* objectives: surrogates that (i) correlate with the routing primitives that underlie Theorem 1 and Theorem 2, and (ii) admit efficient estimation and incremental evaluation for candidate edges.

From conductance to estimable surrogates. A convenient bridge between conductance and efficiently computable quantities is the (normalized) Laplacian spectrum. Let $\mathcal{L} = I - D^{-1/2}AD^{-1/2}$ denote the normalized Laplacian of a graph H, with eigenvalues $0 = \lambda_1(\mathcal{L}) \leq \lambda_2(\mathcal{L}) \leq \cdots \leq \lambda_n(\mathcal{L}) \leq 2$. By Cheeger's inequality,

$$\frac{\lambda_2(\mathcal{L})}{2} \;\leq\; \phi(H) \;\leq\; \sqrt{2\lambda_2(\mathcal{L})}. \tag{9}$$

Thus, *increasing* $\lambda_2(\mathcal{L}_{G'})$ serves as a principled proxy for increasing $\phi(G')$. Moreover, $\lambda_2(\mathcal{L})$ can be approximated to constant relative accuracy by standard iterative methods (power iteration / Lanczos) using only matrix–vector products with A and D, in time $\widetilde{O}(m)$ for sparse graphs when a constant number of iterations suffices for a coarse estimate. This motivates the first proxy:

$$\mathrm{Score}_{\mathrm{gap}}(G') \;:=\; \widehat{\lambda_2(\mathcal{L}_{G'})},$$

where $\widehat{\lambda_2}$ is any efficiently computed lower bound (so that improvements are certified).

However, directly re-estimating λ_2 after each tentative edge addition is still expensive if we consider many candidates. We therefore introduce a second proxy based on electrical notions, which yields cheap *marginal-gain* estimates for single-edge additions.

Effective resistance and Dirichlet energy as routing proxies. Let L be the (combinatorial) Laplacian of H and L^+ its Moore–Penrose pseudoinverse. For vertices u, v, the *effective resistance* is

$$\mathsf{R}_H(u,v) \;:=\; (\mathbf{e}_u - \mathbf{e}_v)^\top L^+(\mathbf{e}_u - \mathbf{e}_v).$$

Effective resistance simultaneously controls random-walk commute times and quantifies how "well connected" two vertices are through multiple disjoint paths. In particular, large $\mathsf{R}_H(u,v)$ is a reliable indicator of a bottleneck separating u and v; conversely, adding an edge (u, v) is most valuable when it bridges a high-resistance pair. This suggests the resistance-based objective

$$\mathrm{Score}_{\mathrm{res}}(G, E_+) \ := \sum_{(u,v) \in E_+} \mathsf{R}_G(u, v), \tag{10}$$

or, more conservatively for sequential addition, the greedy choice of (u, v) maximizing $\mathsf{R}_H(u, v)$ in the *current* graph H.

The justification is that adding a unit-weight edge (u, v) performs a rank-one update $L \mapsto L + \mathbf{b}_{uv}\mathbf{b}_{uv}^\top$ where $\mathbf{b}_{uv} = \mathbf{e}_u - \mathbf{e}_v$, and standard Sherman–Morrison identities imply that global electrical quantities (e.g. $\mathrm{Tr}(L^+)$, sometimes called the Kirchhoff index) decrease by an amount monotone in $\mathsf{R}_H(u, v)$. While we will not require an exact formula, the qualitative consequence is sufficient for our purposes: *high-resistance edges have large certified marginal benefit* for improving global connectivity measures that upper bound mixing time and support low-congestion flows.

Crucially, many effective resistances can be estimated in near-linear time: by Johnson–Lindenstrauss reduction, it suffices to compute an embedding $\mathbf{z}_u \in \mathbb{R}^{O(\log n)}$ such that $\|\mathbf{z}_u - \mathbf{z}_v\|_2^2 \approx \mathsf{R}_H(u, v)$ for all queried pairs, and such embeddings can be produced using $\widetilde{O}(\log n)$ Laplacian solves. In a centralized preprocessing regime this yields $\widetilde{O}(m)$–$\widetilde{O}(m \log n)$ time for a large batch of queries; in restricted regimes one may approximate resistances for sampled pairs via short random walks (using the relationship between resistance and commute time) as a cheaper but less uniform alternative.

Two simple rewiring heuristics. We instantiate the above proxies in two preprocessors that are intentionally simple and compatible with both global and per-node budgets.

(H1) Fiedler-cut edge injection. We compute an approximate second eigenvector $\mathbf{f}$ of $\mathcal{L}_G$ (or, equivalently, a low-Rayleigh-quotient vector orthogonal to $\mathbf{1}$). We then sort vertices by $\mathbf{f}_u$ and identify a sparse cut via a sweep procedure. Given a cut S, we add edges across $(S, V \setminus S)$ to increase $|\partial S|$ subject to the budgets: under a global budget we add a near-uniform random matching between S and $V \setminus S$ of size $\min\{R, |S|, |V \setminus S|\}$; under a per-node budget we cap the number of new incident edges by r (e.g. by sampling at most r partners on the other side for each vertex). This heuristic is directly aligned with the bottleneck accounting in (5): it targets the cut most responsible for the small value of λ_2.

(H2) Resistance-greedy (or resistance-sampled) overlay. We generate a candidate set $\mathcal{C}$ of vertex pairs, either by uniform sampling, by sampling endpoints of random walks, or by using extreme pairs in the resistance embedding. We then add edges from $\mathcal{C}$ in decreasing order of estimated $\mathsf{R}_G(u, v)$, skipping candidates that would violate the per-node cap. In the sampled variant, we pick (u, v) with probability proportional to the estimate of $\mathsf{R}_G(u, v)$ and add

R edges i.i.d. (again with per-node capping). This is the additive analogue of leverage-score sampling in spectral sparsification, with the roles reversed: rather than preserving a spectrum with few edges, we aim to *improve* connectivity by adding edges where the current graph is electrically "thin."

Special-case guarantees and interpretation. On the canonical hard families from Section 7, these heuristics recover the correct linear-budget scaling and, in several cases, yield constant conductance with $R = \Theta(n)$ and $r = O(1)$.

Two-cluster graphs. Let G be the union of two constant-degree expanders A, B connected by a single edge. For $u \in A$ and $v \in B$, $\mathsf{R}_G(u, v) = \Theta(1)$ because any unit flow must traverse the unique inter-cluster bottleneck, whereas for u, v within the same expander we have $\mathsf{R}_G(u, v) = \widetilde{O}(1/n)$ (more precisely $O(1/n)$ up to constants depending on the expander). Consequently, resistance-greedy overwhelmingly selects inter-cluster edges until the cut is repaired, at which point cross-cluster resistances drop. In parallel, the Fiedler vector essentially separates A and B, so (H1) adds edges exactly across the planted partition. After inserting $\Theta(n)$ cross edges in a sufficiently dispersed manner (e.g. a random matching), standard expansion arguments imply $\phi(G') = \Omega(1)$ with high probability.

Paths and grids. On P_n, $\mathsf{R}_{P_n}(u, v) = \Theta(\mathrm{dist}(u, v))$, so resistance-greedy prefers long chords. Adding a random perfect matching (or, more generally, a constant-degree random regular overlay) yields an expander with high probability; since adding edges cannot decrease conductance below that of the overlay, $G' = P_n \cup H$ has $\phi(G') = \Omega(1)$ for such overlays. While the grid has a more delicate resistance metric (with R scaling like $\Theta(\log n)$ at long distances), the same principle applies: resistance-based selection produces long-range connections that destroy planar separators. In practice we implement this by adding a bounded-degree random regular overlay, which has a direct $r = O(1)$ realization and an $\widetilde{O}(n)$ construction time; the grid edges then serve only to decrease shortest-path distances further and do not obstruct expansion.

Why these objectives match the refinement task. Our upper bounds for one-step WL refinement reduce the task to global ranking/sorting of local-type tokens, whose round complexity on G' depends on the existence of low-congestion routing schemes. Both $\lambda_2(\mathcal{L}_{G'})$ and resistance-based global indices are classical predictors of routability: they control mixing, support oblivious routing bounds in various models, and correlate strongly with the empirical performance of expander-routing subroutines. Hence, although conductance itself is the cleanest parameter in theorems, the proxies above are better aligned with a feasible preprocessor: they are efficiently estimable, admit incremental edge-selection rules, and on the hard families they provably drive the instance into the constant-conductance regime exactly when the linear-budget barrier permits it.

In the next section we evaluate these rewiring objectives experimentally, measuring (i) proxy improvement, (ii) realized WL color recovery, and (iii) the observed depth needed by routing-based refinement as a function of preprocessing cost.

9 Experiments

We include an experimental evaluation whose purpose is not to tune a learning system, but to validate the algorithmic claims that motivate our preprocessing objectives: (i) increasing expansion proxies should systematically reduce the depth required by routing-based refinement, and (ii) the linear-budget barrier for raising conductance on standard hard families should manifest as a visible phase transition in observed round complexity.

Protocol. For each input graph $G = (V, E)$ we generate an initial coloring $x \in [p(n)]^n$ (unless the dataset provides node attributes), apply a rewiring preprocessor to obtain $G' = (V, E \cup E_+)$ under either a global budget $|E_+| \leq R$ or a per-node cap $\deg_{E_+}(u) \leq r$, and then run a fixed refinement routine on G' with width parameter w. We compare against a centralized ground truth for one-step refinement, computed by explicitly forming $\text{type}_G(u)$ for all u and applying perfect hashing to obtain $y = \text{WL1}(G, x)$. This isolates the *distributed computability* effect of rewiring from any modeling confounders.

Refinement implementation and measured depth. We instantiate the message-passing stage with a packet-routing/sorting pipeline consistent with the requirements of Theorem 1: each node forms its local key

$$k_u := \big(x_u,\ \{\{x_v : v \in N_{G'}(u)\}\}\big),$$

encodes it into $O(\Delta' \log n)$ bits, and participates in a global ranking procedure that assigns identical ranks to identical keys. Concretely, we implement token routing by repeated randomized permutation routing with congestion control; the measured depth d is the number of synchronous RL-CONGEST rounds until all tokens reach their destinations and a total order of keys is reconstructed. Since routing time depends sharply on expansion, this definition of d is sensitive to $\phi(G')$ (or, more realistically, to the routability properties that correlate with our proxies) while keeping all other components fixed. For the fingerprint variant (Theorem 2), we replace ranking by a one-sided hash $h_u = H(\text{seed}, \text{Enc}(k_u))$ with $b = O(\log(n/\varepsilon))$ bits; here d includes the time to disseminate the seed if it is not assumed as a shared feature.

Metrics. We report three families of measurements.

1. *WL color recovery.* In the exact variant we check whether the distributed output $\widehat{y}$ satisfies $\widehat{y}_u = \widehat{y}_v \Leftrightarrow \mathrm{type}_G(u) = \mathrm{type}_G(v)$ by comparing against the centralized types; we summarize by the fraction of nodes whose label matches the canonical ground truth up to a global renaming (equivalently, by the induced partition error). In the fingerprint variant we estimate the empirical collision rate among unequal types and verify the one-sided condition (equal types never disagree) by construction.

2. *Depth/round complexity.* We record the observed d as above, and also normalize by the packing factor Δ'/w to separate degree blow-up from expansion effects.

3. *Proxy/capacity correlation.* We compute (or lower bound) $\widehat{\lambda_2(\mathcal{L}_{G'})}$, a sweep-cut estimate $\widehat{\phi}(G')$ derived from the same spectral vector, and resistance-based global indices (e.g. sampled averages of $\mathsf{R}_{G'}(u,v)$). We then evaluate the correlation between d and the predictors suggested by the theorems, e.g.

$$\widehat{d}_\phi \;:=\; \frac{\Delta'}{w} \cdot \frac{\mathrm{polylog}(n)}{\widehat{\phi}(G')}, \qquad \widehat{d}_\lambda \;:=\; \frac{\Delta'}{w} \cdot \frac{\mathrm{polylog}(n)}{\sqrt{\widehat{\lambda_2(\mathcal{L}_{G'})}}},$$

   where the polylog(n) factor is fixed across runs and serves only to compare scaling.

We additionally report preprocessing cost (wall-clock time and asymptotic surrogate counters such as the number of Laplacian solves, candidate-pair queries, or random-walk samples) to expose the trade-off between preprocessor strength and feasibility.

Synthetic families (controlled). We evaluate on families where expansion bottlenecks are known and budgets admit a clean interpretation: paths P_n, $\sqrt{n}\times\sqrt{n}$ grids, two-cluster graphs formed by joining two constant-degree expanders with a single edge, lollipop/barbell variants, and random geometric graphs (low conductance due to locality). For each family we vary n and sweep R (or r) across a range from $o(n)$ to $\Theta(n)$ while holding w fixed. The principal observation we seek is a transition from polynomial depth to polylogarithmic depth once rewiring reaches the regime where $\widehat{\phi}(G')$ becomes bounded away from 0. In these controlled settings we also verify the necessity aspect empirically: for $R \ll n$ the proxies remain small and routing depth grows proportionally to $1/\widehat{\phi}(G')$, consistent with the cut-capacity intuition behind Theorem 3, whereas for $R = \Theta(n)$ the preprocessors of Section 8 rapidly drive $\widehat{\lambda_2}$ upward and collapse the observed depth.

Real graphs (ecological validity). We complement synthetic instances with standard graph benchmarks spanning disparate mixing behavior: road networks (very low conductance), citation/coauthorship graphs (moderate expansion with heavy tails), and web/social networks (often higher conductance but with community structure). For each graph we consider two input-feature regimes: (i) *uninformative* colors $x_u \equiv 1$ to stress the need for global coordination even for simple WL keys, and (ii) *given* categorical attributes where present. We compare multiple preprocessors under the same budget: the objectives from Section 8, a purely random overlay of the same size, and degree-biased heuristics (e.g. connecting high-degree vertices) that increase diameter less reliably. The real-graph evaluation is structured to test whether the proxy improvements translate into reductions in d beyond what can be explained by shrinking shortest-path distances alone.

Ablations on preprocessing cost and information. To make the preprocessing–computability trade-off explicit, we ablate (a) the number of iterations used to approximate spectral quantities, (b) the size $|\mathcal{C}|$ of the candidate set for resistance-based selection, (c) the choice of candidate generation (uniform pairs versus random-walk endpoints), and (d) the enforcement mechanism for per-node caps (hard rejection versus soft reweighting). We also ablate whether a global random seed is assumed as an initial shared feature or must be disseminated over G'; this directly affects the fingerprinting depth in low-conductance regimes and clarifies when shared randomness should be treated as an external resource rather than "free."

Summary of expected empirical patterns. Across families we expect (and in our implementation we observe) that (i) improvements in $\widehat{\lambda_2}$ and decreases in resistance-based indices track reductions in routing depth more robustly than graph-diameter statistics, (ii) under fixed $r = O(1)$ the degree blow-up Δ' remains controlled and thus the factor Δ'/w does not dominate the depth once conductance improves, and (iii) the most pronounced gains occur precisely on graphs with a single dominant bottleneck (paths, grids, two-cluster), where both the spectral and resistance proxies concentrate their marginal gain on repairing that bottleneck. These measurements serve as evidence that the rewiring objectives are aligned with the refinement primitive that drives our theorems, rather than being generic graph-improvement heuristics.

Reproducibility notes. All experiments are parameterized by (w, R) or (w, r) and report n, m, Δ' alongside the measured d and proxy estimates; randomization is controlled by a fixed set of public seeds. We emphasize that the experimental goal is comparative and scaling-oriented: the relevant question is whether, under identical message width and refinement logic, the rewiring choices that improve expansion proxies are the same choices that reduce the observed depth needed for WL-type canonicalization.

10 Discussion and Open Problems

We conclude by isolating several gaps between the upper and lower bounds, and by outlining extensions and modeling choices that appear technically consequential rather than cosmetic.

Tightness and the role of polylogarithmic factors. Our upper bounds for one-step refinement on a rewired graph G' with $\phi(G') \geq \phi$ take the form

$$d \;\leq\; \widetilde{O}\Big(\frac{\Delta'}{w} \cdot T_{\text{route}}(n, \phi)\Big),$$

where T_{route} is instantiated via a particular routing/sorting primitive. The lower bound (Theorem 3) shows that, for collision-free refinement, there exist bounded-degree graphs of conductance $\Theta(\phi)$ requiring

$$d \;=\; \Omega\Big(\frac{1}{\phi} \cdot \frac{m}{w \log n}\Big).$$

Even restricting to $\Delta' = O(1)$ and $m = \Theta(n)$, there remains a conceptual gap: the lower bound scales as $\Omega((1/\phi) \cdot n/(w \log n))$, while the upper bound inherits whatever ϕ-dependence is implicit in $T_{\text{route}}(n, \phi)$ and may additionally hide $\text{polylog}(n)$ factors. We view two questions as central.

(i) Optimal ϕ-dependence for routing-based canonicalization. Existing routing bounds on conductance-ϕ graphs are often stated in terms of mixing time $\tau_{\text{mix}}(G')$ or spectral gap, and are typically tight only up to polylogarithmic factors. Determining whether WL-type ranking intrinsically requires $\Omega(1/\phi)$ rounds (even on constant-degree graphs) or whether the dependence can be improved to $\text{polylog}(1/\phi)$ for some relaxed notion of correctness would sharpen the phase-transition picture suggested by Theorem 5.

(ii) Tightness of the cut-capacity lower bound for collision-free refinement. Theorem 3 derives an $\Omega(1/\phi)$ factor via a sparse cut. It is open whether there exists a matching upper bound for collision-free refinement that scales as $\widetilde{O}((1/\phi) \cdot m/(w \log n))$ on every graph, or whether the existential family is genuinely harder than worst-case routing on conductance-ϕ graphs. Put differently, we do not yet know whether collision-free *global consistency* of relabeling is strictly harder than moving $\Theta(n)$ tokens with constant congestion on the same topology.

Exact versus approximate refinement: what is the correct objective? A recurring modeling choice is whether we demand perfect hashing semantics (exact WL1) or allow bounded error via fingerprinting (Theorem 2). Our fingerprint guarantee is one-sided: equal types always agree, and unequal types collide with probability at most ε. This is natural when WL is used as a refinement operator inside a larger randomized computation, but it raises several open directions.

(i) Necessity of global ranking for exact WL1. Our exact algorithm reduces refinement to global ranking of keys. An alternative is to attempt *distributed perfect hashing* without explicit sorting, e.g., by constructing a distributed dictionary keyed by $\mathrm{Enc}(k_u)$, or by incremental resolution of collisions. We do not know whether such approaches can asymptotically beat sorting-based implementations on high-conductance graphs under the same w, nor whether they can avoid the $\Omega(m)$-bits-across-a-cut phenomenon in Theorem 3.

(ii) Two-sided error and relaxed semantics. In many learning-motivated settings, it may suffice that unequal types collide with small probability and equal types collide with probability close to 1 (rather than identically 1). This changes the information-theoretic landscape because nodes need not construct a canonical encoding of their full multiset; they can use sketches whose equality is only probabilistic in both directions. Characterizing the minimal depth for such *sketch-only* semantics, and the extent to which conductance still governs it, remains open.

(iii) Multi-round WL and error accumulation. Our discussion and experiments focus on a single WL refinement step. Iterating t steps introduces dependencies: either one must disseminate new randomness per step, or reuse a seed and control correlations. Establishing a clean composition theorem of the form "t steps at total error δ in $\widetilde{O}(t \cdot \Delta'/w \cdot \mathrm{polylog}(n))$ rounds" requires care even on expanders, and appears nontrivial on low-conductance graphs where seed dissemination itself may dominate.

Directed graphs, weighted graphs, and heterogeneity of bandwidth. Our statements are phrased for undirected, unweighted graphs. Extending them requires choosing which analytic quantity replaces conductance. For directed graphs, a standard option is to work with the lazy random walk on a strongly connected directed graph and define an analogue of ϕ using stationary flow across cuts; however, routing primitives and cut-capacity arguments must then account for edge directions, which can create bottlenecks not present in the symmetrized graph. For weighted graphs, the natural generalization replaces edge counts by capacities and degrees by weighted degrees in $\mathrm{vol}(\cdot)$, but the WL key formation step also becomes ambiguous: should the multiset contain neighbor colors with multiplicities proportional to weights, or should weights be treated as separate features? We expect that a coherent extension is possible by defining $\mathrm{type}_G(u)$ to include weight-annotated neighbor colors and by measuring cut capacity via total boundary weight, but we do not currently have tight routing-to-ranking reductions stated in that language. A further practical complication is heterogeneous bandwidth: if the per-edge width is not uniform (e.g., depends on weight/capacity), then Δ'/w should be replaced by a congestion parameter derived from edge capacities, and it is unclear whether our packing argument remains sharp.

Virtual nodes, tokenization, and what counts as rewiring. Our preprocessing model adds edges E_+ under either a global budget R or per-node budget r. Many applied architectures also permit *virtual nodes* (a new hub connected to all original nodes) or auxiliary "register" nodes used for aggregation. From our perspective, such operations are rewiring with a very particular structure: a virtual node adds n edges incident to a single node, which violates $r = O(1)$ but may respect a global $R = \Theta(n)$ budget. This highlights that the *distribution* of added degree, not only the edge count, matters for Δ' and thus for the Δ'/w packing factor. An open problem is to formalize a preprocessing model that charges separately for (i) total added edges, (ii) maximum added degree, and (iii) introduction of new nodes, and then to characterize which combinations permit polylogarithmic-depth refinement without rendering the model vacuous. Relatedly, even without adding new nodes, one may attempt to simulate "virtual edges" by routing tokens along existing paths; this is precisely where conductance and mixing enter, and it would be useful to quantify when explicit rewiring offers a strict advantage over such emulation under fixed d.

Shared randomness as a resource. The fingerprint variant assumes access to a seed, either as an initial shared feature or disseminated over G'. In RL-CONGEST, treating a public seed as "free" is a strong assumption on sparse, low-conductance graphs: disseminating even $O(\log n)$ bits may take $\Omega(1/\phi)$ rounds if the seed must cross a sparse cut. This suggests two distinct models: (i) *private coins only*, where any correlation must be induced by communication, and (ii) *public coins*, where common randomness is an external resource. The separation is not merely definitional: one can plausibly obtain fingerprinting in polylogarithmic depth on expanders with either model, but on low-conductance graphs public coins might circumvent parts of the lower-bound construction if the hard instance relies on uncertainty that public randomness would remove. Determining whether Theorem 3 (or a variant) holds under public coins for collision-free refinement, and identifying the exact point at which shared randomness changes the depth regime for bounded-error tasks, remain open.

Implications for benchmark design. Finally, we extract a methodological lesson. Many graph-learning benchmarks emphasize average-case performance on fixed datasets, where the underlying graphs often have moderate conductance or contain high-degree hubs that facilitate rapid information flow. Our results suggest that such benchmarks may underrepresent instances where depth is fundamentally limited by expansion. If the goal is to evaluate preprocessing schemes or message-passing architectures as *distributed algorithms*, then benchmarks should include families with controlled bottlenecks and explicit budget constraints, and should report not only accuracy but also the achieved $(\Delta', \phi(G'))$ and the number of rounds required by a fixed canonicalization primitive (such as WL1). In particular, it is informative to include instances where $\phi(G)$ is tunably small (paths, grids, barbell/lollipop, geometric graphs) and to sweep R (or r) across the $\Theta(n)$ threshold suggested by Theorem 4, thereby making the predicted phase transition in computability observable rather than incidental.