Local-Access Graph Foundation Models: Sublinear Few-Shot Adaptation with Prompt Tokens

Liz Lemma Future Detective

January 20, 2026

Abstract

Few-shot learning on graphs has recently shifted from meta-learning to pre-training and prompt-based, parameter-efficient adaptation. Yet most work implicitly assumes full-graph access at downstream time, which is unrealistic on 2026-era web-scale graphs where privacy, latency, and storage constraints prevent loading |V| and |E|. Motivated by the survey's emphasis on large-scale graphs and parameter-efficient adaptation, we introduce a local-access computational model for few-shot node/edge prediction where the learner only queries bounded-radius neighborhoods. We propose LA-Prompt, which uses a pre-trained subgraph tokenizer and frozen graph encoder, and adapts with a tiny prompt learned from K labeled examples. We provide tight learning-theoretic guarantees for r-local tasks: sample complexity matching lower bounds up to logarithmic factors and inference time depending only on the retrieved neighborhood size. Finally, we show sharp limitations: for global graph properties, any local-access algorithm with sublinear oracle calls cannot succeed in the worst case. Experiments (recommended) on large real graphs measure latency–accuracy tradeoffs under strict neighborhood budgets and verify the theory's locality assumptions.

Table of Contents

4. 4. Tokenization and Frozen Encoders: definition of pre-trained neighborhood tokenizers; what is assumed vs what can be learned; discussion of practical instantiations (pooling, hashing, learned coarsening).

5. 5. Algorithm (LA-Prompt): prompt-only adaptation and local inference; variants for node classification and link prediction; caching and batching under neighborhood queries.

6. 6. Upper Bounds for r-Local Tasks: formal assumptions, excess risk bound, and per-query complexity independent of |V|,|E|.

7. 7. Matching Lower Bounds (Learning): minimax/sample complexity lower bound in the same hypothesis class; show near-tightness of LA-Prompt.

8. 8. Limits of Local Access (Oracle Lower Bounds): impossibility for global/non-local tasks; reductions from property testing / communication complexity; guidance on when LA-FSL is appropriate.

9. 9. Experimental Protocol (Recommended): large-graph latency–accuracy benchmarking; strict oracle budget emulation; ablations on r, token count m, prompt size P, caching; stress tests on tail/cold-start subpopulations.

10. 10. Discussion and Extensions: dynamic/heterogeneous graphs, adaptive radius, retrieval-augmented neighborhoods, and integrating prompts with lightweight adapters.

11. 11. Conclusion: what is provably achievable with local access and small prompts; implications for graph foundation models.

1 Introduction

Graphs that matter operationally in 2026—web and commerce interaction graphs, enterprise knowledge graphs, financial transaction networks, and multi-relational user–item graphs—are routinely too large, too dynamic, or too access-restricted to admit the algorithmic assumption that one can load the entire adjacency structure and perform global computation. Even when storage is feasible, organizational constraints (privacy, multi-tenant deployment, regulatory separation, or API-only access) often restrict a downstream learner to retrieving small neighborhoods around a limited set of queried entities. In such regimes, the classical distinction between "training time" and "test time" blurs: inference itself may require fetching neighborhood information, and any adaptation to a new task must be expressed as a small computation over a small collection of locally retrieved subgraphs.

We therefore take local access as a primitive: the learner interacts with a target graph through an oracle that returns a rooted induced neighborhood of bounded radius. This access model is not merely a convenience for analysis; it matches the dominant engineering pattern for massive graphs, where feature generation and candidate retrieval are inherently neighborhood-based and cached, and where the cost metric is not the number of floating point operations but the number and size of graph fetches. From this perspective, the fundamental question is not "Can a graph neural network fit the task given the entire graph?" but rather "Which tasks can be solved, and with what sample and query complexity, when every unit of information arrives as a bounded-radius neighborhood around instances of interest?"

At the same time, the past few years have clarified that pre-training can dramatically reduce label requirements in domains where labeled data are scarce but structure is abundant. Graph representation learning has followed the same trajectory as language and vision: large-scale self-supervised objectives on unlabeled graphs can produce feature extractors that transfer to many downstream tasks. Yet, in graph settings, transfer is complicated by heterogeneity across domains (different node types, attribute vocabularies, and relation semantics) and by the locality constraints above. A downstream practitioner may be permitted to compute embeddings for queried nodes, but not to re-train or even fine-tune a large backbone that was pre-trained elsewhere. Thus, parameter-efficient adaptation—prompting, adapters, low-rank updates—is not merely fashionable; it is often the only admissible mechanism for personalization, continual updates, or task switching under a strict deployment budget.

Our aim is to unify these considerations in a single formal framework: (i) we allow an offline pre-training stage with full access to external unlabeled graphs, producing a tokenizer and encoder that map rooted neighborhoods to fixed-dimensional embeddings; (ii) we require that downstream learning on a new graph proceeds only by local oracle queries; and (iii) we constrain

adaptation to a small number of trainable parameters, which we interpret as a prompt. The resulting model is deliberately austere: the backbone is frozen, the downstream algorithm cannot traverse the graph arbitrarily, and the only labeled data are a few support examples for the task at hand. This combination captures the practical pattern "retrieve neighborhood $\rightarrow$ embed $\rightarrow$ lightly adapt $\rightarrow$ predict," while making explicit the resource trade-offs among neighborhood radius, number of oracle calls, number of labeled examples, and prompt capacity.

Two themes motivate our analysis. First, locality is both an opportunity and a limitation. It is an opportunity because many economically important prediction problems are plausibly r-local: node classification based on profile and nearby interactions, link scoring based on shared neighborhoods, anomaly detection based on local motifs, and so on. In such cases, oracle access is information-theoretically sufficient, and the learning problem reduces to identifying a predictor over frozen local embeddings from few labels. It is a limitation because not all graph properties are local: connectivity, expansion, long-range community structure, and other global predicates cannot, in general, be inferred from a sublinear number of bounded-radius samples. Any framework that purports to address learning on massive graphs must therefore articulate where positive results are possible and where impossibility results intervene.

Second, the success of pre-training and prompting suggests a separation of concerns: representation is learned once at scale, and adaptation is performed cheaply and locally. In our setting, the tokenizer and encoder summarize each oracle-retrieved neighborhood into a fixed number of tokens and a fixed embedding. Downstream learning then operates in this induced feature space. This makes the dependence on the size of the target graph disappear from the computational complexity, except through the random size of retrieved neighborhoods, and it isolates the role of the prompt: the prompt is responsible for aligning the pre-trained features with the downstream labeling function using only K labeled support instances. When the downstream task is compatible with the representation—formally, when an accurate predictor lies in a bounded-norm linear class over the frozen embeddings—we can expect label efficiency comparable to standard linear prediction, with explicit dependence on embedding dimension rather than on $|V|$ or $|E|$.

This perspective speaks directly to two challenges that recur in surveys of modern graph learning systems. The first is the *large-scale challenge*: the target graph is too large to process globally, so algorithms must be instance-local and must bound the number of neighborhood expansions. The second is the *structure-scarcity challenge*: labels are scarce or expensive, while unlabeled structure is plentiful. Our framework addresses both by construction: it makes locality an explicit constraint (rather than a heuristic) and treats labeled data as a few-shot resource. Moreover, by incorporating

a prompt budget, we account for realistic deployment constraints in which one may tune only a small number of parameters per task, per client, or per time window.

Finally, we emphasize that adopting local access as a core assumption demands matching lower bounds. If the local neighborhoods do not contain the information needed to solve a task, no amount of prompting or pre-training can overcome the information bottleneck at downstream time. The proper comparison class is therefore not "all graph predictors," but the class of predictors that can be expressed as functions of rooted neighborhoods of bounded radius (after pre-training). In what follows, we make this comparison precise: we present an instance-local prompting algorithm and analyze its excess risk under locality and bounded-norm hypotheses, and we complement this with lower bounds that delineate when such guarantees are tight and when they are impossible. The next section situates these ideas relative to existing work on few-shot graph learning, large-scale neighborhood sampling, parameter-efficient adaptation, and property-testing indistinguishability phenomena.

2 Related Work

Few-shot learning on graphs has been studied under several paradigms that differ primarily in what is assumed transferable across tasks and what computational access is available to the target graph. A first line follows meta-learning: one trains across many episodic tasks so that adaptation from a small support set is fast at test time, using either optimization-based updates (e.g., MAML-style methods) or metric-based rules (e.g., prototypes and matching) **??**. In graph settings, these ideas appear in few-shot node classification and relation prediction, where tasks correspond to label subsets, relation types, or domains, and adaptation is implemented by a small number of gradient steps on a GNN backbone or on a task-specific head **??**. While effective in moderate-scale benchmarks, many meta-learning methods still assume that (at adaptation time) the learner can repeatedly traverse the target graph and backpropagate through the full model, which conflicts with deployment constraints in which the backbone is frozen and graph access is mediated by restricted neighborhood queries.

A second paradigm is pre-training followed by light-weight transfer. Self-supervised and weakly supervised graph representation learning has produced a large family of objectives—contrastive alignment across augmentations, context prediction, mutual-information surrogates, and masked attribute/edge reconstruction—that yield embeddings reusable across downstream tasks **???**. This "pre-train then probe" philosophy is conceptually aligned with our setting, in that the downstream learner operates on frozen features and the statistical question becomes whether the downstream labels

are predictable from those features. However, much of the pre-training literature is formulated either with full access to each training graph or in an "in-memory" regime where mini-batches can sample arbitrary subgraphs using stored adjacency, whereas our downstream phase treats the target graph as an oracle-access object and measures cost in neighborhood fetches.

Prompting and other parameter-efficient adaptation methods provide a third axis. In language and vision, prompt tuning, adapters, and low-rank updates (LoRA) enable task adaptation by training a small number of parameters while keeping a large backbone fixed **???**. Graph analogues have recently emerged: one can introduce learnable "virtual" nodes or tokens, add prompt vectors to node representations, or learn small adapter modules inserted between GNN layers **??**. These methods are typically motivated by the same resource constraints that motivate our formulation—multi-task deployment, per-client personalization, and limits on fine-tuning time—but they are often evaluated in settings where the entire graph (or a substantial subgraph) is available during adaptation. Our contribution is not a new prompting mechanism per se, but an access model that makes explicit that the prompt must be trainable from a few labeled, instance-local neighborhoods, with oracle calls counted as a primary resource.

Large-scale graph learning has also developed techniques that approximate global training with local computation. Neighborhood sampling and mini-batch methods such as GraphSAGE-style sampling, FastGCN/LADIES, GraphSAINT, and cluster-based batching reduce the cost of training deep GNNs by restricting message passing to sampled neighborhoods **?????**. Production recommenders further combine sampling with retrieval and caching layers (e.g., PinSAGE) to compute embeddings on demand **?**. These works share with our approach the premise that only a bounded portion of the graph can be touched per training example, and they provide practical estimators for stochastic gradients. The distinction is that sampling methods usually presume direct access to adjacency (to sample neighbors, to precompute random walks, or to build clusters), whereas our downstream learner is restricted to the neighborhoods returned by an external oracle, a model closer to API-bound graph stores and privacy-separated deployments.

The neighborhood oracle viewpoint is also related to classical models of local computation and distributed graph algorithms. In the LOCAL model, an algorithm at a node observes its radius-r neighborhood after r synchronous rounds; many impossibility results show that certain global predicates cannot be decided from bounded-radius views **?**. In property testing, one studies sublinear-time algorithms that query local neighborhoods (or incidences) to distinguish a property from being far from it; here, indistinguishability constructions yield sharp lower bounds for tasks such as connectivity, expansion, and partition properties under local queries **??**. These results motivate our negative statements: when two graph families have (approximately) identical distributions over rooted r-neighborhoods,

no downstream algorithm constrained to such views can reliably separate them, regardless of how powerful the frozen encoder may be. Our framework imports this indistinguishability principle into a learning setting, where the goal is to predict labels rather than to decide a single property.

Few-shot graph learning additionally intersects with work on inductive transfer across graphs and domains. Methods that train on multiple graphs and generalize to unseen graphs often rely on structural regularities shared across domains, sometimes using relational inductive biases, subgraph encoders, or graph-level contrastive objectives **?**. While these approaches address domain shift, they typically do not formalize a strict downstream access budget, and they allow adaptation procedures that implicitly depend on the ability to traverse beyond queried instances. Our model isolates a stricter regime: the algorithm may only see neighborhoods of the support and query instances (up to explicitly budgeted auxiliary queries), which is natural when predictions are served at query time and any additional exploration is costly or disallowed.

Finally, our learning-theoretic stance relates to analyses of linear probing and representation quality. A common formalization is: given frozen features, a downstream task is easy if a low-complexity predictor (often linear with bounded norm) achieves small risk, and then generalization depends primarily on the feature dimension and the norm bound **?**. We adopt this viewpoint in the presence of local graph access and a prompt budget: the statistical difficulty is governed by the complexity of predictors over frozen neighborhood embeddings, while the algorithmic difficulty is governed by the number and size of oracle-retrieved neighborhoods. In summary, our work sits at the intersection of (i) graph few-shot transfer, (ii) parameter-efficient adaptation, (iii) large-scale neighborhood-based computation, and (iv) local-access lower bounds from property testing, with the goal of making the tradeoffs among labels, prompt capacity, and oracle access explicit.

3 Model and Problem Formulation (LA-FSL)

We formalize a downstream learning regime in which the target graph is too large, too private, or too remotely stored to be processed as an explicit adjacency structure. The learner is instead granted *local* access to the graph through an oracle, and must adapt to a task from a small labeled support set while modifying only a small number of trainable parameters. This section specifies the access model, the resource budgets, and the families of tasks we aim to capture.

Target graph and oracle access. Let the target be a graph $G = (V, E, X)$ with node features $X \in \mathbb{R}^{|V| \times d}$ (edge features are omitted unless stated). Fix

a radius $r \in \mathbb{N}$. Our only mechanism for observing G is a neighborhood oracle

$$O_G(v, r) \;=\; \text{the rooted induced } r\text{-hop neighborhood around } v,$$

which returns the subgraph induced by nodes at graph distance at most r from v, together with their features and an explicit root identifier. We denote a returned neighborhood by $G_v^{(r)} := O_G(v, r)$ and write $|E_r|$ for the number of edges in such a neighborhood (which may depend on v and may be treated as a random variable under a distribution over query points). The downstream computational cost will be measured as a function of the number of oracle calls and the sizes of the returned neighborhoods, and must not scale with $|V|$ or $|E|$ except through such local statistics.

Instances and task types. A downstream task T is specified by labeled *instances* of one of two canonical forms. For *node prediction*, an instance is a node $o = v \in V$ with label $y(v) \in \mathcal{Y}$ (e.g., a class). For *edge/link prediction*, an instance is an ordered pair $o = (u, v) \in V \times V$ with label $y(u, v) \in \mathcal{Y}$ indicating presence, type, or some relational attribute. In the edge case, we assume access to $O_G(u, r)$ and $O_G(v, r)$ (or, equivalently, a joint oracle that returns a rooted neighborhood around the pair); the analysis is insensitive to the choice provided the algorithm is charged for each neighborhood retrieval. We emphasize that the oracle returns *induced* neighborhoods: the learner cannot request arbitrary subsets of neighbors, nor can it traverse beyond radius r without issuing further counted oracle calls.

Support/query protocol and learning objective. The downstream input consists of a support set

$$S = \{(o_i, y_i)\}_{i=1}^{K},$$

where each o_i is a node or node-pair as above, and a set Q of query instances on which we must predict. We view instances as drawn from a distribution $\mathcal{D}$ induced by sampling nodes or pairs in G (and then revealing the associated labels through an unknown labeling function). The goal is to output a predictor $\hat{y}(o)$ achieving low expected risk

$$\mathbb{E}_{o \sim \mathcal{D}}\big[\ell(\hat{y}(o), y(o))\big],$$

for a specified loss ℓ (typically convex and 1-Lipschitz in its prediction argument when we later state generalization bounds). The support set size K is the statistical resource, and we treat the query set Q as unlabeled at adaptation time.

Two-phase representation and frozen backbone. We assume an offline pre-training phase on external unlabeled graphs (full access permitted there) that produces a frozen representation mechanism. Concretely, the downstream learner is given a fixed mapping from an oracle neighborhood to a finite-dimensional vector representation. For the purposes of the present section we denote this mapping abstractly by

$$\Phi(\,\cdot\,) \;:\; O_G(v,r) \mapsto \mathbb{R}^D,$$

and defer its construction and practical instantiations to the next section. The defining constraint is that Φ is *frozen* downstream: its parameters cannot be updated using the support set.

Prompt/adaptation budget. Adaptation to a downstream task is performed by training a *parameter-efficient* module (a prompt) with at most P real-valued degrees of freedom. Formally, we consider a family of predictors of the form

$$\hat{y}(o) \;=\; g_\varphi\big(\Phi(\mathsf{neigh}(o))\big),$$

where $\varphi \in \mathbb{R}^P$ are the only trainable parameters available at downstream time, and $\mathsf{neigh}(o)$ denotes the oracle neighborhood(s) required to represent o (one neighborhood for node tasks, two for edge tasks, up to bookkeeping conventions). The map g_φ may be a linear probe, a small MLP, or a prompt mechanism that modifies intermediate activations, but its trainable footprint is capped by P. This budget models deployment regimes in which per-task or per-client fine-tuning must be fast, cheap to store, and safe to perform without modifying a shared backbone.

Neighborhood and oracle-call budgets. In addition to limiting trainable parameters, we restrict graph access. A downstream algorithm may only invoke $O_G(\cdot, r)$ on (i) nodes appearing in $S \cup Q$, and (ii) an explicitly bounded set of auxiliary nodes, with each invocation counted toward an oracle budget q. We allow memoization: repeated calls on the same root may be cached and charged once. Computation performed after receiving a neighborhood must be polynomial in the returned subgraph size, and any overall complexity bounds must depend on r and local size measures (e.g., $|E_r|$) rather than on global graph size. This constraint rules out adaptation procedures that require repeated global passes over G or that rely on precomputing graph-wide data structures.

Local task families and the r-locality hypothesis. Our positive results require a compatibility condition between the task and the access model. We call a labeling function *r-local* if, for node tasks, $y(v)$ is a function only of the rooted neighborhood $O_G(v, r)$, and similarly for edge tasks if

$y(u, v)$ is determined by $O_G(u, r)$ and $O_G(v, r)$ (or an equivalent local view). This hypothesis captures many settings in which labels depend on bounded-range patterns (features, motifs, or short-range relational context), and it is the minimal assumption under which oracle access at radius r can be information-theoretically sufficient. In contrast, tasks depending on global graph properties (e.g., connectivity or membership in a giant component) are not r-local for fixed r and will be subject to the lower bounds we later state.

Reference predictor class over frozen features. To separate *representation quality* from *few-shot learnability*, we benchmark against predictors that are simple functions of frozen embeddings. A canonical reference is the bounded-norm linear class

$$\mathcal{H}_B \;=\; \Big\{\, o \mapsto \langle w, \Phi(\mathsf{neigh}(o))\rangle \;:\; \|w\|_2 \le B \,\Big\},$$

possibly composed with a fixed link function for classification. Our learning objective can then be stated as achieving small excess risk relative to $\inf_{h\in\mathcal{H}_B} \mathbb{E}[\ell(h(o), y(o))]$ using only K labeled examples, q oracle calls, and P trainable parameters.

What is and is not allowed downstream. We stress three invariants of the LA-FSL model. First, the algorithm cannot inspect G beyond queried neighborhoods; in particular, it cannot sample random nodes unless such sampling is itself implemented by counted oracle queries. Second, the backbone representation Φ is immutable, so the only path to task specialization is through the prompt parameters φ (and any associated small head). Third, resource bounds must be *instance-local*: per-query prediction should require only a constant number of oracle calls and computation scaling with the returned neighborhood size, ensuring feasibility when $|V|$ and $|E|$ are massive.

This completes the downstream problem definition. In the next section we instantiate Φ via pre-trained neighborhood tokenizers and frozen encoders, and we discuss which aspects are assumed fixed versus learned during pre-training.

4 Tokenization and Frozen Encoders

We now instantiate the frozen neighborhood representation map Φ used downstream. Rather than treating Φ as a monolithic black box, we factor it into (i) a *neighborhood tokenizer* that converts an r-hop rooted neighborhood into a fixed number of token vectors, and (ii) a *frozen encoder* that aggregates these tokens into a task-agnostic embedding. This factorization makes

the locality and resource constraints explicit: the tokenizer and encoder operate only on oracle outputs, and the token budget fixes the downstream compute independent of the global size of G.

Neighborhood tokenizer. Let $G_v^{(r)} := O_G(v, r)$ denote the rooted induced r-hop neighborhood around a node v. A neighborhood tokenizer is a map

$$\tau_\psi : G_v^{(r)} \longmapsto Z_v \in \mathbb{R}^{m \times p},$$

where m is a fixed token count and p is the token dimension. The parameters ψ are learned (or chosen) during pre-training and then frozen downstream. The requirement that m be fixed is not merely cosmetic: it decouples the cost of encoding from the possibly heavy-tailed size of $G_v^{(r)}$. The tokenizer may compress neighborhoods of varying size by pooling, coarsening, truncation with padding, or any permutation-invariant summarization of the rooted subgraph; the only hard constraint is that τ_ψ be computable in time polynomial in $|V(G_v^{(r)})| + |E(G_v^{(r)})|$.

Since $G_v^{(r)}$ is a rooted object, τ_ψ may (and typically should) allocate one distinguished token to the root. Concretely, one may view Z_v as containing a "root token" capturing the features of v and its immediate relational context, plus $m-1$ auxiliary tokens describing the remainder of the neighborhood at increasing hop distance or at increasing coarseness. Rooting also resolves the usual ambiguity of graph permutation symmetry: while the neighborhood is unordered, the distinguished root provides a canonical reference point for relative structural features (e.g., hop distance to the root, directionality if present, or role features).

Frozen encoder. Given tokens $Z_v \in \mathbb{R}^{m \times p}$, the frozen encoder produces an embedding

$$h_v \;=\; f_\theta(Z_v) \in \mathbb{R}^D, \qquad \text{and we set} \qquad \Phi(G_v^{(r)}) \;:=\; f_\theta(\tau_\psi(G_v^{(r)})).$$

The encoder parameters θ are learned during offline pre-training and remain fixed at downstream time. The encoder may be a Transformer operating on m tokens, an MLP applied to pooled tokens, or any architecture with predictable complexity as a function of m and p. Importantly, we do *not* assume the downstream learner can re-run message passing on the full neighborhood with trainable weights; all trainable adaptation is deferred to the prompt module in the next section.

What is assumed fixed versus learned. In the downstream phase, the pair (τ_ψ, f_θ) is immutable. In particular, neither the tokenization scheme nor the encoder weights are updated on the support set, and their parameter counts do not contribute to the downstream prompt budget P. Our

theoretical statements treat Φ as a fixed feature map; the substantive modeling assumption is that, for the target task family of interest, there exists a low-complexity predictor (often linear with bounded norm) over these frozen embeddings that approximates the optimal r-local decision rule. Said differently, pre-training is responsible for producing features in which r-local structure becomes linearly (or simply) predictable; few-shot adaptation is responsible only for selecting a task-specific decision boundary within that fixed feature space.

Offline, by contrast, we allow substantial flexibility. The tokenizer and encoder can be trained by any self-supervised or weakly supervised objective on external graphs, including contrastive neighborhood discrimination, masked attribute prediction, or predictive coding between overlapping neighborhoods. The analysis in later sections does not depend on the particular pre-training loss, only on the induced hypothesis class over $\Phi(G_v^{(r)})$.

Practical instantiations of tokenizers. Several concrete choices of τ_ψ satisfy the oracle-based locality constraint while yielding fixed-size token sequences.

Pooling-based tokenization uses hand-designed or lightly parameterized summaries. For example, one may allocate one token per hop distance $\ell \in \{0, \dots, r\}$, with entries given by pooled statistics of node features at that hop (means, variances, or learned linear projections), and optionally include structural statistics such as degree histograms or counts of small motifs within each shell. This yields $m = r + 1$ (or a small multiple thereof) and offers strong robustness to neighborhood size variation, at the cost of potentially discarding fine-grained relational information.

Hashing- or color-refinement tokenization converts rooted neighborhoods into multisets of discrete identifiers. A typical approach is to run a small number of Weisfeiler–Lehman-style refinement steps within $G_v^{(r)}$, hash the resulting colors (optionally combined with hop distance to the root), and then form tokens by aggregating embeddings of the hashed identifiers. This can be implemented efficiently, is naturally permutation-invariant, and can be tuned to trade off expressivity against token budget by controlling how many hashed buckets are retained.

Learned coarsening tokenization learns a soft partition of the neighborhood into m clusters ("supernodes") and pools node representations within each cluster. Concretely, one may compute preliminary node states by a small *frozen* message-passing network inside τ_ψ, then predict an assignment matrix $S \in \mathbb{R}^{|V(G_v^{(r)})| \times m}$ and output token vectors $Z_v = S^\top H$ for node state matrix H. In this view, tokens are learned subgraph summaries whose number is fixed by design; the coarsening can be trained during pre-training to preserve information relevant for generic structural prediction.

Truncation-and-padding tokenization linearizes the neighborhood by a

canonical traversal rooted at v (e.g., BFS with deterministic tie-breaking), selects up to a fixed number of visited nodes/edges, and encodes them as tokens with positional or hop-distance features, padding when the neighborhood is smaller than the budget. This approach can retain fine details but may be sensitive to the traversal rule; it is best paired with data augmentation during pre-training to encourage invariance.

Edge tasks and pair representations. For link prediction or edge labeling, we will typically compute $h_u = \Phi(O_G(u, r))$ and $h_v = \Phi(O_G(v, r))$ via two oracle calls, and then combine them (e.g., concatenation, bilinear scoring, or an additional interaction token) before applying the prompt-conditioned predictor. Our abstraction permits either scheme: the essential point is that each component representation is derived from a bounded-radius oracle view with fixed token count.

With $\Phi = f_\theta \circ \tau_\psi$ fixed in this way, downstream learning reduces to fitting a small prompt-conditioned predictor on top of embeddings computed from oracle neighborhoods, which we make explicit in the next section.

5 LA-Prompt: Prompt-Only Adaptation Under Local Oracle Access

With the frozen neighborhood representation map $\Phi = f_\theta \circ \tau_\psi$ in place, we now specify the downstream procedure that performs task adaptation using only (i) r-hop oracle access and (ii) a bounded prompt budget P. The guiding constraint is that every computation at downstream time must be instance-local: for a node instance we may inspect only $O_G(v, r)$, and for an edge instance only the corresponding endpoint neighborhoods (or a joint neighborhood oracle when available). In particular, we do not assume access to global adjacency lists, full-graph message passing, or any operation whose cost scales with $|V|$ or $|E|$.

Problem interface and embedding extraction. For each labeled support instance we first compute a frozen embedding by a single pass through the tokenizer and encoder. In the node case, the support set is $S = \{(v_i, y_i)\}_{i=1}^K$ and we form

$$G_i \;:=\; O_G(v_i, r), \qquad Z_i \;:=\; \tau_\psi(G_i) \in \mathbb{R}^{m \times p}, \qquad h_i \;:=\; f_\theta(Z_i) \in \mathbb{R}^D.$$

These steps are deterministic given the oracle output and the frozen pair (τ_ψ, f_θ), and thus the downstream learner may treat $\{(h_i, y_i)\}_{i=1}^K$ as the effective training set. The same pipeline is used at inference time for each query node $v \in Q$.

Prompt-only adaptation objective. Adaptation is implemented by a small trainable module p_ϕ with parameter vector $\phi \in \mathbb{R}^P$ (optionally together with a similarly small prediction head), while the backbone (ψ, θ) remains fixed. Concretely, we fit ϕ by empirical risk minimization on the support embeddings:

$$\widehat{\phi} \;\in\; \arg\min_{\phi \in \mathbb{R}^P} \frac{1}{K} \sum_{i=1}^{K} \ell(g_\phi(h_i), y_i) \;+\; \lambda \mathcal{R}(\phi).$$

Here g_ϕ denotes the prompt-conditioned predictor, ℓ is the task loss (e.g., logistic for classification or squared for regression), and $\mathcal{R}$ is a regularizer (e.g., $\|\phi\|_2^2$) used to stabilize few-shot adaptation. In the simplest instantiation, g_ϕ is a linear probe whose weights are the prompt parameters, $g_\phi(h) = \langle w(\phi), h \rangle$, so that $P = D$ (or $P = DC$ for C-way classification). More generally, p_ϕ may modulate the token sequence or the embedding in a low-dimensional way, after which a fixed (or also small) readout is applied. Typical examples that respect the P-budget include: (i) additive prompt tokens inserted into the encoder input (learned only downstream while the encoder is frozen), (ii) feature-wise affine modulation $h \mapsto \alpha(\phi) \odot h + \beta(\phi)$ with $P = 2D$ or low-rank parametrization, and (iii) a low-rank adapter $h \mapsto h + U(\phi)V(\phi)^\top h$ with rank chosen to keep P small. Our analysis in later sections treats p_ϕ abstractly, requiring only that the number of trained degrees of freedom is P and that all oracle-dependent computation passes through Φ.

Local inference. Given $\widehat{\phi}$, prediction for a query node $v \in Q$ is obtained by a single oracle call and a single frozen forward pass:

$$\widehat{y}(v) \;:=\; g_{\widehat{\phi}}(f_\theta(\tau_\psi(O_G(v, r)))) \,.$$

The salient point is that the representation cost depends on r (through the size of the returned neighborhood) and on the fixed token budget m, but is independent of the global graph size. Thus per-query inference is well-defined even when G is massive, provided the oracle can return local neighborhoods.

Variant: link prediction and edge labeling. For edge tasks the instance is an ordered or unordered pair (u, v). Under the local-access constraint we represent the pair using only oracle views rooted at the endpoints:

$$h_u \;:=\; \Phi(O_G(u, r)), \qquad h_v \;:=\; \Phi(O_G(v, r)).$$

We then define a pair feature map $\Gamma : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}^{D'}$ and predict via $\widehat{y}(u, v) = g_\phi(\Gamma(h_u, h_v))$. Standard symmetric choices include $\Gamma(h_u, h_v) = [h_u \| h_v \| h_u \odot h_v \| |h_u - h_v|]$, or a bilinear score $\langle A h_u, h_v \rangle$ where A is prompt-parameterized with P degrees of freedom (e.g., diagonal or low-rank). If the

interface provides a joint oracle $O_G((u, v), r)$ returning an induced neighborhood around both endpoints, we may instead tokenize and encode that joint rooted object and proceed exactly as in the node case; our framework accommodates both, with the endpoint-based variant requiring at most two oracle calls per edge instance.

Caching and oracle-call accounting. Because oracle calls are the dominant non-differentiable interaction with the target graph, we make the access pattern explicit. In the node setting, the naive downstream transcript uses $K + |Q|$ calls, one per distinct node in $S \cup Q$. In the edge setting, a naive implementation uses $2(K + |Q|)$ calls, but in practice many pairs share endpoints. We therefore memoize either the oracle outputs $O_G(v, r)$ or, more compactly, the embeddings $h_v = \Phi(O_G(v, r))$. With caching, the oracle-call count equals the number of *distinct* nodes appearing as endpoints across support and query instances, and repeated occurrences incur only constant-time table lookup. This observation is essential in regimes such as link prediction where evaluating many candidate neighbors for a single node is natural: the marginal cost per additional candidate edge can be reduced to the cost of combining already-cached embeddings.

Batching and predictable downstream compute. Although oracle neighborhoods have variable raw size, the fixed token count m yields a uniform tensor shape for the encoder input. Consequently, once we have retrieved and tokenized a batch of neighborhoods, we can stack the token matrices in $\mathbb{R}^{b \times m \times p}$ and apply the frozen encoder in a single batched call. In the downstream phase we therefore separate (i) neighborhood retrieval, which may be asynchronous and irregular, from (ii) representation learning compute, which is regular and amenable to acceleration. The same batching applies to adaptation: prompt fitting is performed over the fixed-size embeddings $\{h_i\}$ and thus reduces to standard optimization whose cost depends on (K, P, D) rather than on $|V|$ or $|E|$.

In summary, LA-Prompt reduces few-shot learning on a massive, locally accessible graph to (a) a bounded number of oracle neighborhood queries, (b) a frozen feature extraction step with fixed token budget, and (c) optimization over at most P trainable parameters. We next formalize the conditions under which this pipeline provably learns any r-local task that is linear (with bounded norm) in the frozen representation, and we derive excess-risk and per-query complexity bounds that do not scale with the size of G.

6 Upper Bounds for r-Local Tasks

We now state conditions under which prompt-only adaptation on top of the frozen neighborhood map $\Phi := f_\theta \circ \tau_\psi$ achieves small excess risk using only K

labeled support instances, and we make explicit that both the statistical and computational guarantees are independent of the global graph size ($|V|, |E|$).

Statistical model and r-locality. We analyze node tasks for clarity; the edge case follows by applying the same argument to a pairwise feature map $\Gamma(h_u, h_v)$ as described previously. Let $\mathcal{D}$ denote the distribution over labeled examples induced by sampling a node v (according to the task interface) and observing the label $y(v)$ together with its rooted neighborhood $O_G(v, r)$. The downstream learner observes a support set $S = \{(v_i, y_i)\}_{i=1}^K$ where $(v_i, y_i) \sim \mathcal{D}$ i.i.d. (or, more generally, satisfying the standard concentration conditions needed for uniform convergence). The defining structural assumption is that the Bayes rule depends only on the r-hop rooted induced neighborhood:

$$y(v) \;=\; h^\star\big(O_G(v, r)\big) \qquad \text{for some measurable } h^\star. \tag{1}$$

Since the downstream algorithm may access G only through $O_G(\cdot, r)$ on queried roots, (1) is precisely the condition under which local access is information-theoretically sufficient.

Linear realizability over frozen embeddings. Write the frozen embedding of a rooted neighborhood as

$$\mathbf{h}(v) \;:=\; \Phi\big(O_G(v, r)\big) \;=\; f_\theta\Big(\tau_\psi\big(O_G(v, r)\big)\Big) \in \mathbb{R}^D.$$

We assume $\|\mathbf{h}(v)\|_2 \le R$ almost surely under $\mathcal{D}$ (this can be enforced by normalization in the encoder). The reference hypothesis class is the bounded-norm linear family

$$\mathcal{H}_B \;:=\; \big\{v \mapsto \langle w, \mathbf{h}(v)\rangle :\; \|w\|_2 \le B\big\}.$$

In the simplest instantiation of LA-Prompt, the prompt parameters directly encode w (so $P = D$ for binary prediction, or $P = DC$ for C-way one-vs-rest), while more structured prompts restrict w to a lower-dimensional subset. Our bound is stated relative to $\mathcal{H}_B$; any prompt parameterization that can represent (or approximate) the risk minimizer in $\mathcal{H}_B$ inherits the same guarantee up to approximation error.

Loss assumptions and risk. Let $\ell(\hat{y}, y)$ be convex and 1-Lipschitz in $\hat{y}$ for each fixed y (e.g., logistic loss). Define the population and empirical risks

$$L(w) \;:=\; \mathbb{E}_{(v,y)\sim\mathcal{D}}\big[\ell(\langle w, \mathbf{h}(v)\rangle, y)\big], \qquad \widehat{L}_S(w) \;:=\; \frac{1}{K}\sum_{i=1}^{K} \ell(\langle w, \mathbf{h}(v_i)\rangle, y_i).$$

We take $\widehat{w}$ to be the empirical risk minimizer (or a regularized minimizer, e.g. ridge/logistic with $\lambda\|w\|_2^2$); the standard stability/optimization issues are orthogonal to locality and are handled by the usual convex analysis.

Theorem 6.1 (Excess-risk bound for r-local linear tasks)**.** *Assume* (1), $\|\mathbf{h}(v)\|_2 \le R$ *almost surely, and* ℓ *is convex and* 1*-Lipschitz in prediction. Then with probability at least* $1-\delta$ *over the draw of* S*, the empirical minimizer* $\widehat{w} \in \arg\min_{\|w\|_2 \le B} \widehat{L}_S(w)$ *satisfies*

$$L(\widehat{w}) - \inf_{\|w\|_2 \le B} L(w) \;\le\; c\,\frac{BR}{\sqrt{K}} \;+\; c\,BR\sqrt{\frac{\log(1/\delta)}{K}},$$

for a universal constant $c > 0$*. In particular, it suffices to take*

$$K \;=\; \widetilde{O}\Big(\frac{B^2R^2 + \log(1/\delta)}{\varepsilon^2}\Big)$$

to guarantee excess risk at most ε*.*

Proof sketch. We treat the frozen map Φ as part of the data-generating process: conditioned on each oracle output, $\mathbf{h}(v)$ is deterministic. The Rademacher complexity of $\mathcal{H}_B$ over K samples is at most $BR/\sqrt{K}$, since $\sup_{\|w\|\le B} \sum_i \sigma_i \langle w, \mathbf{h}(v_i)\rangle = B\|\sum_i \sigma_i \mathbf{h}(v_i)\|_2$ and $\mathbb{E}\|\sum_i \sigma_i \mathbf{h}(v_i)\|_2 \le R\sqrt{K}$. By the contraction inequality for 1-Lipschitz losses and standard symmetrization, uniform deviation $|L(w) - \widehat{L}_S(w)|$ is controlled at the same scale, yielding the stated high-probability excess risk bound for empirical minimization. If the prompt module p_ϕ is used, one either (i) analyzes the induced predictor class $\{g_\phi \circ \Phi\}$ directly via its capacity (typically scaling with P under norm constraints), or (ii) reduces to $\mathcal{H}_B$ when p_ϕ is expressive enough to realize the optimal linear w (and accounts for any mismatch as an additive approximation term).

Per-query complexity and independence from $(|V|, |E|)$**.** The statistical guarantee above is agnostic to the size of the target graph; it depends only on K and the norm bounds. Computationally, prediction for a query node v requires exactly one oracle call to obtain $O_G(v, r)$, followed by tokenization and a frozen forward pass:

$$O_G(v,r) \;\xrightarrow{\;\tau_\psi\;}\; Z \in \mathbb{R}^{m\times p} \;\xrightarrow{\;f_\theta\;}\; \mathbf{h}(v) \in \mathbb{R}^D \;\xrightarrow{\;g_{\widehat{\phi}}\;}\; \widehat{y}(v).$$

Let $|E_r| := |E(O_G(v,r))|$ denote the number of edges in the returned neighborhood. The downstream-time cost of tokenization is $\mathrm{Tok}(r) = \mathrm{poly}(|E_r|, d, m)$ by construction of τ_ψ, and the cost of the frozen encoder is $\mathrm{Enc}(r)$, e.g. $\widetilde{O}(L \cdot |E_r|)$ for an L-layer message-passing backbone restricted to the neighborhood, or $\widetilde{O}(m^2)$ for attention over m tokens. The prompt/readout adds $O(P)$ time. Crucially, no step requires iterating over V or E; hence per-query time is

$$\widetilde{O}\big(\mathrm{Tok}(r) + \mathrm{Enc}(r) + P\big),$$

which is independent of $|V|$ and $|E|$ except through the local neighborhood statistics governed by r and the oracle.

7 Matching Lower Bounds for Learning

We complement the preceding upper bound with a minimax lower bound showing that, even under the same r-local linear realizability assumptions, no learner can in general improve the ε^{-2} dependence (nor the $1/\sqrt{K}$ rate) using only K labels. Importantly, the lower bound holds for *any* algorithm—local-access or otherwise—and therefore reflects an intrinsic statistical limitation of learning bounded-norm linear predictors from few labeled examples, rather than a deficiency of the oracle interface.

A reduction to a one-dimensional r-local family. Fix any radius $r \geq 0$. Consider a family of target graphs in which every node is isolated (no edges) and carries a single scalar feature; then $O_G(v, r)$ reveals exactly that scalar feature (together with the root identifier), and hence the task is trivially r-local. Composing with the frozen map $\Phi = f_\theta \circ \tau_\psi$ only strengthens the learner; thus, for a lower bound it suffices to consider the case in which the embedding returned to the downstream learner is simply $\mathbf{h}(v) = x(v) \in \mathbb{R}$ with $|x(v)| \leq R$. Since $\mathcal{H}_B$ contains all one-dimensional predictors $x \mapsto wx$ with $|w| \leq B$, any lower bound for this one-dimensional subfamily transfers immediately to the full D-dimensional class.

Loss choice. We instantiate the bound with the hinge loss $\ell(z, y) = \max\{0, 1 - yz\}$, which is convex and 1-Lipschitz in z. (Any other convex 1-Lipschitz loss admits an analogous two-point construction; hinge makes the algebra transparent because the risk is linear in a neighborhood of the origin.)

Theorem 7.1 (Minimax lower bound for r-local bounded-norm linear prediction)**.** *Let $\ell(z, y) = \max\{0, 1 - yz\}$. Fix $B, R > 0$ and set $R' := \min\{R, 1/B\}$ so that $BR' \leq 1$. For each $\Delta \in (0, 1/4]$, define two distributions $\mathcal{D}_+$ and $\mathcal{D}_-$ over labeled examples (x, y) by*

$$x \equiv R', \qquad \mathbb{P}_{\mathcal{D}_\pm}(y = +1) = \frac{1}{2} \pm \Delta, \quad \mathbb{P}_{\mathcal{D}_\pm}(y = -1) = \frac{1}{2} \mp \Delta.$$

Then:

1. *The population risk minimizers over $[-B, B]$ satisfy $w_+^\star = +B$ for $\mathcal{D}_+$ and $w_-^\star = -B$ for $\mathcal{D}_-$.*
2. *For any (possibly randomized) learning algorithm A that maps a sample S of K i.i.d. examples to an output $\widehat{w} = \mathsf{A}(S) \in [-B, B]$, there exists a choice of sign $\sigma \in \{+, -\}$ such that*

$$\mathbb{P}_{S \sim \mathcal{D}_\sigma^K}\big(L_\sigma(\widehat{w}) - L_\sigma(w_\sigma^\star) \;\geq\; 2\Delta\, BR'\big) \;\geq\; \frac{1}{4},$$

where $L_\sigma(w) := \mathbb{E}_{(x,y)\sim\mathcal{D}_\sigma}[\ell(wx,y)]$. *Consequently, achieving excess risk at most* ε *with probability at least* $3/4$ *uniformly over this family requires*

$$K \;=\; \Omega\!\left(\frac{B^2R'^2}{\varepsilon^2}\right) \;=\; \Omega\!\left(\frac{\min\{B^2R^2,1\}}{\varepsilon^2}\right).$$

Proof sketch. We first compute the risk gap under $\mathcal{D}_+$; the $\mathcal{D}_-$ case is symmetric. Since $BR' \le 1$, for any $w \in [-B,B]$ we have $|w|R' \le 1$ and hence both margins $1 - wR'$ and $1 + wR'$ are nonnegative. Therefore

$$L_+(w) = \Big(\tfrac{1}{2}+\Delta\Big)\big(1-wR'\big) + \Big(\tfrac{1}{2}-\Delta\Big)\big(1+wR'\big) = 1 - 2\Delta\, wR'.$$

Thus $L_+(w)$ is strictly decreasing in w, so the constrained minimizer is $w_+^\star = B$, and for any $w \le 0$ we have

$$L_+(w) - L_+(w_+^\star) = \big(1 - 2\Delta\, wR'\big) - \big(1 - 2\Delta\, BR'\big) \ge 2\Delta\, BR'.$$

Hence it suffices to show that, for some choice of sign $\sigma \in \{+,-\}$, the algorithm outputs the wrong sign with probability at least $1/4$.

To this end we apply Le Cam's two-point method. The K-sample distributions $\mathcal{D}_+^K$ and $\mathcal{D}_-^K$ differ only in the bias of y, so their total variation distance is controlled by their KL divergence:

$$\mathrm{TV}\big(\mathcal{D}_+^K, \mathcal{D}_-^K\big) \le \sqrt{\tfrac{1}{2}\mathrm{KL}\big(\mathcal{D}_+^K \,\|\, \mathcal{D}_-^K\big)} = \sqrt{\tfrac{K}{2}\,\mathrm{KL}\Big(\mathrm{Bern}\big(\tfrac{1}{2}+\Delta\big)\,\Big\|\,\mathrm{Bern}\big(\tfrac{1}{2}-\Delta\big)\Big)}.$$

For $\Delta \le 1/4$ the Bernoulli KL is $\Theta(\Delta^2)$, so choosing $\Delta = c/\sqrt{K}$ for a sufficiently small universal constant c ensures $\mathrm{TV}(\mathcal{D}_+^K, \mathcal{D}_-^K) \le 1/2$. Le Cam's inequality then implies that any decision rule (in particular, the sign of $\widehat{w}$ produced by A) errs with probability at least $1/4$ on one of the two cases. Combining this with the explicit risk gap above yields the stated lower bound, and setting $\Delta = \Theta(\varepsilon/(BR'))$ gives $K = \Omega(B^2R'^2/\varepsilon^2)$.

Implication for LA-Prompt. Because the construction is r-local and can be realized on graphs with trivial neighborhoods, the lower bound applies a fortiori in our oracle model. Thus, up to logarithmic factors and normalization choices (often $R = 1$ by design), the ε^{-2} label requirement exhibited by prompt-only adaptation over frozen embeddings is minimax-optimal in the worst case within this r-local bounded-norm linear regime.

8 Limits of Local Access (Oracle Lower Bounds)

We now delineate a complementary limitation of the local-access model which is orthogonal to the statistical lower bound of Section 7. There, even full

access to G cannot circumvent the ε^{-2} label dependence for bounded-norm linear prediction. Here, by contrast, we allow unlimited labels and computation, but restrict the downstream learner to oracle access $O_G(\cdot, r)$ with fixed radius r and a bounded number q of oracle calls. We show that for tasks whose target depends on *global* properties of G (or, more generally, on information not determined by the distribution of rooted r-hop neighborhoods), no algorithm making $q = o(|V|)$ local queries can succeed with constant probability in the worst case.

A generic indistinguishability principle. Fix any (possibly randomized, adaptive) oracle algorithm A which, on input a query instance o (node or edge) and support set S, makes at most q calls to $O_G(\cdot, r)$ and returns a prediction $\widehat{y}$. The interaction between A and the oracle induces a transcript random variable

$$\mathsf{Tr} = \big((a_1, O_G(a_1, r)), \ldots, (a_t, O_G(a_t, r))\big), \qquad t \le q,$$

where each a_j is the jth adaptively chosen oracle argument (typically a node in $S \cup Q$, or an auxiliary node if permitted by the budget). Crucially, conditioned on the internal randomness of A, the output $\widehat{y}$ is a measurable function of Tr alone. Hence, if two graph distributions $\mathcal{G}_0, \mathcal{G}_1$ are such that the induced distributions of Tr are identical (or sufficiently close in total variation), then the output distributions of *any* local-access algorithm coincide (or are close), and consequently no such algorithm can reliably distinguish the two cases. This is the familiar "black-box oracle" viewpoint: when the oracle answers are coupled to look the same, the algorithm has no additional handle on the underlying global structure.

Property-testing reductions: globally different, locally identical. The standard way to instantiate the above principle is to construct two graph families which are far apart in the target property but locally indistinguishable up to radius r on most roots. One canonical example is connectivity (or, more robustly, expansion). Let $\mathcal{G}_1$ be a distribution over connected d-regular expanders on $n = |V|$ vertices, and let $\mathcal{G}_0$ be a distribution obtained by taking the disjoint union of two independent d-regular expanders on $n/2$ vertices each. The property "G is connected" differs deterministically between $\mathcal{G}_1$ and $\mathcal{G}_0$, yet for any fixed $r = O(1)$ the rooted r-hop neighborhood of a uniformly random vertex is (with high probability) a d-ary tree of depth r under *both* distributions. Intuitively, the presence or absence of a single macroscopic cut is not witnessed inside a bounded ball around a typical vertex. More refined variants replace connectivity by "far from connected" in the sense of property testing, which yields robustness to small perturbations and makes the indistinguishability stable under the addition of a sublinear number of adversarial edges.

By Yao's minimax principle, once we exhibit such $\mathcal{G}_0, \mathcal{G}_1$ whose local views coincide on all but an $o(1)$ fraction of vertices, any (possibly randomized) local-access algorithm that probes only $q = o(n)$ roots will, with high probability, see only "typical" neighborhoods and thus obtain (essentially) the same transcript distribution under both cases. Therefore, its best achievable success probability at deciding the global bit (and hence predicting any label encoding that bit) is bounded away from 1. This yields an oracle lower bound of the form: for any fixed r, there exist global tasks for which $q = \Omega(|V|)$ oracle calls are necessary to achieve error below (say) $1/3$.

Communication-complexity reductions: hidden bits dispersed across components. A second, complementary template encodes a hard instance of *indexing* or related one-way communication problems into far-separated regions of the graph. For example, we may take G to be a disjoint union of M components, each component encoding one bit of a hidden string $b \in \{0,1\}^M$ in a way that is detectable from an r-hop view *only if* the algorithm queries a node inside that component. The downstream query (or the label for a designated root) depends on a particular coordinate b_j. Any local-access algorithm that makes q oracle calls can inspect at most q components (up to constant factors), and therefore learns negligible information about b_j when j is uniformly random unless $q = \Omega(M)$. Translating back to graphs with $|V| = \Theta(M)$ (by keeping component sizes constant in n) yields again a linear-in-$|V|$ oracle requirement. Unlike the property-testing construction, this argument cleanly separates *information acquisition* (which components were visited) from *computation* (arbitrary post-processing of the visited neighborhoods), underscoring that the bottleneck is the oracle interface itself.

Consequences for LA-FSL: when the model is appropriate. These oracle lower bounds justify why our positive guarantees must assume r-locality (or an approximation thereof). In practice, LA-FSL is well matched to settings where the label of a node or edge is determined primarily by a bounded-radius ego-network together with its features—e.g., homophilous node classification, motif-based roles, or link prediction driven by shared neighborhoods. Conversely, if the task depends on graph-wide structure (community membership defined by a global partition, connectivity to a rare hub, centrality measures requiring long-range paths, existence of a planted cut, etc.), then either (i) the required radius r must grow with $|V|$ (defeating the locality premise), or (ii) the number of oracle calls must scale linearly in $|V|$ in the worst case.

Practical guidance. The preceding discussion suggests a simple diagnostic: if two graphs (or two regions of the same graph) can be made to have essentially the same distribution of rooted r-hop neighborhoods while differing

in the target label, then no method that only consumes $O_G(\cdot, r)$ outputs—including any prompt-based adaptation on frozen representations—can be expected to succeed uniformly. Accordingly, when deploying LA-FSL one should either select tasks with an explicit locality rationale, or enlarge the access model (e.g., allow random-walk sampling, limited global sketches, or additional side information) and account for the corresponding query budget. This perspective also clarifies the role of caching: memoization reduces redundant calls for repeated roots, but it does not alter the worst-case necessity of probing many *distinct* regions when the signal is globally distributed.

9 Experimental Protocol (Recommended)

We recommend an evaluation protocol that makes the local-access constraints explicit and reports accuracy–latency tradeoffs under a controlled oracle budget. Since the downstream phase in our model only observes G through calls to $O_G(\cdot, r)$, the experiment should emulate this interface as strictly as possible: all methods must obtain graph structure and features solely via oracle queries, and any additional access (e.g., precomputed global statistics) must be counted as an augmentation to the model and reported separately.

Tasks, splits, and reporting. We instantiate the downstream task T either as node classification with instances $o = v$ or as link prediction with instances $o = (u, v)$. We sample a support set $S = \{(o_i, y_i)\}_{i=1}^{K}$ and a query set Q disjoint from S (unless the task definition necessitates overlap). All results should be averaged over multiple random draws of S and Q and over multiple random seeds for the adaptation procedure (when applicable). We report both predictive performance (e.g., accuracy, macro-F1, or AUC as appropriate) and resource usage, in particular (i) the number of oracle calls q, (ii) the number of *distinct* roots queried when caching is used, and (iii) summary statistics of neighborhood sizes such as $\mathbb{E}[|E_r|]$ and high quantiles of $|E_r|$ under the induced distribution of roots in $S \cup Q$.

Strict oracle-budget emulation. To emulate $O_G(v, r)$ on a benchmark graph stored in memory, we implement a wrapper that, given a root v and radius r, returns the rooted induced r-hop subgraph with node features and a canonical root identifier. Critically, the wrapper must not expose adjacency lists beyond the returned neighborhood, and the downstream algorithm must be written against the wrapper interface. For edge tasks, we either query $O_G(u, r)$ and $O_G(v, r)$ separately (counting two calls) or define an explicit joint oracle $O_G((u, v), r)$ if the method requires it (and count one call with appropriately defined output). We recommend logging the entire sequence of oracle arguments to verify adherence to a prescribed call budget.

Latency–accuracy benchmarking on massive graphs. Because the downstream runtime is dominated by neighborhood retrieval and frozen encoding, we evaluate methods under a *latency budget* measured in wall-clock time and under an *oracle budget* measured in calls. We separate the downstream time into (a) neighborhood retrieval time, (b) tokenization time $\mathrm{Tok}(r)$, (c) encoder time $\mathrm{Enc}(r)$, and (d) adaptation/head time. For each configuration, we report a Pareto curve of predictive performance versus (i) per-query time and (ii) total downstream time for processing $S \cup Q$. When possible, we normalize time by the returned neighborhood size to obtain a machine-agnostic proxy, e.g.,

$$\mathsf{cost}(v) \;=\; |E(O_G(v,r))| \quad \text{and} \quad \mathsf{cost}(S) \;=\; \sum_{(v,y)\in S} |E(O_G(v,r))|.$$

This disentangles architectural differences (e.g., attention versus message passing) from dataset-dependent neighborhood growth.

Baselines under the same access model. We recommend comparing LA-Prompt against baselines that respect the oracle interface: (i) frozen embeddings with a linear probe (no prompt), (ii) prompt variants (token-level, activation-level, and head-only) with the same parameter budget P, and (iii) non-pretrained local methods that operate on $O_G(v,r)$ only (e.g., a small GNN trained from scratch on the retrieved neighborhoods). Any baseline that uses full-graph preprocessing (e.g., global normalization, spectral features, label propagation on all of G) should be placed in a separate "augmented access" category, as it violates the intended downstream constraints.

Ablations on the locality and representation budgets. We ablate the principal downstream knobs in a grid that is explicitly tied to the model parameters:

1. *Radius r:* evaluate $r \in \{1, 2, 3, \ldots\}$ up to the point where neighborhood sizes become prohibitive. Report both performance and neighborhood growth, since improvements with larger r may be confounded by increased oracle information.

2. *Token count m:* vary m while keeping the tokenizer architecture fixed. Since m controls the downstream token budget, this isolates whether accuracy gains arise from richer neighborhood summaries or from unrelated training effects.

3. *Prompt size P:* vary P over an order of magnitude (e.g., $P \in \{0, 10^2, 10^3, 10^4\}$) while keeping f_θ frozen. This tests whether few-shot adaptation is primarily limited by statistical signal (K) or by adaptation capacity.

4. *Caching:* compare (a) no caching, (b) perfect memoization of repeated roots, and (c) bounded caches with eviction. Since caching changes the *effective* q for repeated queries, we recommend reporting both total calls and distinct-root calls.

For each ablation, we keep the remaining parameters fixed and report confidence intervals over support/query resampling.

Oracle-budget stress tests. To probe sensitivity to the oracle limitation, we impose hard caps on the number of calls: $q \in \{K,\ K + |Q|,\ 2(K + |Q|)\}$, and (when auxiliary probing is permitted) we allocate a separate auxiliary budget q_{aux}. Methods that adaptively query auxiliary nodes must charge each such query against q_{aux}, and we recommend plotting performance as a function of q_{aux} to reveal whether gains stem from additional information acquisition rather than from better use of the same local views.

Tail and cold-start subpopulations. We recommend reporting stratified results on subpopulations where local-access methods are plausibly brittle: (i) low-degree versus high-degree nodes (degree bins), (ii) rare classes or long-tail labels, (iii) nodes with missing or noisy features, and (iv) newly introduced nodes/edges in temporal splits (cold start). For each stratum, we report not only accuracy but also neighborhood-size statistics, since tail performance may degrade either because the task is harder or because $O_G(\cdot, r)$ returns systematically less informative neighborhoods.

Reproducibility checklist. Finally, we recommend logging: the exact definition of $O_G(\cdot, r)$ (induced versus sampled neighborhoods), the budgets (r, m, P, K, q), the optimizer and regularization used to fit ϕ, and the hardware/runtime settings for latency measurements. Under strict local access, these details are not ancillary; they define the computational problem being solved.

10 Discussion and Extensions

We briefly discuss several extensions of the LA-FSL formalism that preserve the defining constraint—downstream-time access to the target graph only through bounded-radius oracle calls—while broadening the class of graphs and adaptation mechanisms covered by the model. Our intent is not to introduce new primitives gratuitously, but rather to isolate which relaxations are benign (in the sense that they can be accounted for by an explicit budget) and which relaxations fundamentally change the computational problem.

Dynamic graphs and continual downstream adaptation. In many deployments, the target graph evolves over time; we may write $G_t = (V_t, E_t, X_t)$ for discrete times $t = 1, \ldots, T$ and assume oracle access to $O_{G_t}(\cdot, r)$. A natural downstream objective becomes an online risk

$$\sum_{t=1}^{T} \mathbb{E}_{o \sim \mathcal{D}_t} \big[\ell(\hat{y}_t(o), y_t(o)) \big],$$

where $\mathcal{D}_t$ is the induced instance distribution at time t and $\hat{y}_t$ uses prompt parameters ϕ_t updated from a time-varying support stream. If we restrict adaptation at each t to convex prompt objectives over frozen embeddings (e.g., logistic/least-squares heads on $f_\theta(\tau_\psi(\cdot))$), standard online convex optimization yields regret bounds scaling as $\widetilde{O}(\sqrt{T})$ in terms of the prompt parameter norm, provided labels remain r-local with respect to G_t. When the labeler drifts, one may instead bound *dynamic* regret in terms of a variation budget $\sum_t \|\phi^\star_{t+1} - \phi^\star_t\|$, which makes explicit that no local-access method can track arbitrarily fast global shifts without additional supervision. Algorithmically, caching is especially consequential in the dynamic setting: memoized neighborhoods and embeddings become stale when G_t changes, so the cache must be versioned or invalidated; such invalidations should be counted as additional oracle work, as they effectively re-query the environment.

Heterogeneous and attributed graphs. For heterogeneous graphs with node/edge types (and potentially multiple relation sets), the oracle can be taken to return a typed rooted neighborhood subgraph, i.e.,

$$O_G(v, r) \;=\; \big(G[v, r],\ \mathrm{type}_V,\ \mathrm{type}_E,\ X \big),$$

and similarly for edge instances. In this regime, the tokenizer τ_ψ must be interpreted as operating on typed neighborhoods; a minimal modification is to append learned type embeddings to node/edge features before tokenization. The learning-theoretic statements in our main development remain structurally unchanged: once τ_ψ and f_θ are frozen, the downstream learner again reduces to fitting a small predictor on fixed-dimensional embeddings, with sample complexity controlled by the effective embedding dimension and the prompt/head parameterization. What changes is the representational burden placed on pre-training: type-conditional structure must be compressed into m tokens. Empirically, this suggests reporting neighborhood-type statistics (e.g., relation-degree profiles) alongside $|E_r|$, since heterogeneity can inflate local complexity even at fixed radius.

Adaptive radius and variable-cost locality. Our baseline model fixes a radius r globally. In practice, one may wish to choose the radius per instance, trading off information and cost. Formally, we can allow the downstream algorithm to query $O_G(v, r')$ for any $r' \le r_{\max}$, charging a cost that depends on

the returned subgraph size, e.g., $\mathrm{cost}(v, r') = |E(O_G(v, r'))|$. The resulting problem is a constrained decision problem: select $r'(o)$ to minimize risk subject to $\sum_{o \in S \cup Q} \mathrm{cost}(o, r'(o)) \leq C$. If labels are $r_\star$-local for some unknown $r_\star \leq r_{\max}$, a simple doubling strategy (query radii $1, 2, 4, \ldots$ until validation loss stabilizes) identifies a sufficient radius with only a logarithmic overhead in the number of oracle calls for repeated instances (and, with caching, in distinct roots). The nontrivial point is that adaptivity does not violate local access *per se*, but it must be budgeted: larger radii increase both the oracle information and the runtime through the induced growth in $|E_r|$, and thus should be treated as part of the downstream resource vector.

Retrieval-augmented neighborhoods under explicit budgets. A qualitatively different extension is to permit *auxiliary* oracle queries on nodes not present in $S \cup Q$, for the purpose of retrieving additional context. One abstraction is to define, for each instance o, a retrieval rule that selects a set $R(o) \subseteq V$ and augments the representation with the multiset of their neighborhoods:

$$\mathrm{Aug}(o) \;=\; \big\{O_G(u, r) : u \in R(o) \cup \{o\}\big\}.$$

This can be implemented by embedding each retrieved neighborhood via the frozen (τ_ψ, f_θ) pipeline and aggregating (e.g., attention over retrieved embeddings) before applying the prompt/head. Such retrieval strictly increases oracle information, so it must be charged against an auxiliary budget q_{aux}. The benefit is that some tasks that are not strictly r-local may become solvable when a small number of "landmarks" is queried (e.g., tasks depending on membership in a sparse set of communities identifiable from a few representative nodes). However, the lower bounds for global properties remain: if distinguishing cases requires probing a linear fraction of V, retrieval cannot circumvent the $\Omega(|V|)$ oracle barrier, it merely makes the dependence explicit through q_{aux}.

Prompts versus lightweight adapters. Our prompt module p_ϕ was left intentionally broad. In implementations, one often considers lightweight adapters (e.g., low-rank updates to a final projection, token-wise affine shifts, or small bottleneck MLPs) rather than pure "soft prompts." Within our formalism, these are simply alternative parameterizations of the downstream map $g_\phi \circ f_\theta$ with parameter budget P. For convex heads (or convex-in-ϕ prompt parameterizations), the excess-risk bounds follow from uniform convergence for bounded-norm predictors on fixed embeddings. For nonconvex adapters, one typically cannot claim the same worst-case optimization guarantees; nevertheless, if the adapter is low-rank and the effective function class can be controlled (e.g., via norm constraints and Lipschitzness of p_ϕ), the statistical dependence on K still scales with an effective dimension tied

to P rather than to $|V|$ or $|E|$. Conceptually, adapters enlarge approximation power at fixed oracle access, whereas increasing r or permitting retrieval enlarges oracle information; disentangling these effects is essential when interpreting gains.

Scope of what these extensions can and cannot change. All extensions above preserve the central dichotomy established by our theorems: when the target label is determined by a bounded-radius neighborhood statistic and the frozen representation renders the task linearly (or simply) separable, small-K adaptation is possible with downstream complexity governed by local neighborhood size and prompt budget; when the target depends on global structure not compressible into bounded-radius views (even augmented by few auxiliary probes), oracle limitations impose unavoidable failure modes. This perspective leads directly to the concluding question: under which locality and representation assumptions do graph foundation models provide provable downstream utility under strict access constraints?

11 Conclusion

We close by isolating what is *provably* achievable in the local-access few-shot regime and, correspondingly, what claims about "graph foundation models" can be made precise under explicit downstream constraints. The formalism we adopted separates three ingredients that are often conflated in practice: (i) *oracle information*, controlled by the radius r and the number of oracle calls; (ii) *representation*, controlled by the frozen pair (τ_ψ, f_θ) that maps an r-hop rooted neighborhood to a fixed-dimensional embedding; and (iii) *adaptation capacity*, controlled by the prompt (or head) parameter budget P and the number K of labeled support examples. This separation is not merely aesthetic: it determines exactly which improvements can be attributed to better pre-training versus larger downstream access or larger supervised adaptation.

On the positive side, our main message is that few-shot learning on massive graphs is feasible without global graph access provided the task is *local* in an information-theoretic sense and the frozen representation is sufficiently aligned with the relevant neighborhood statistics. Concretely, when the Bayes-optimal labeling rule depends only on $O_G(v, r)$ and, after applying (τ_ψ, f_θ), becomes well-approximated by a bounded-norm linear predictor, LA-Prompt achieves excess risk at most ε using $K = \tilde{O}(D/\varepsilon^2)$ labels (or the corresponding effective dimension for the chosen head/prompt class), with per-instance runtime depending only on the size of the returned neighborhood (e.g., $\tilde{O}(L \cdot |E_r|)$ for message passing, or $\tilde{O}(m^2)$ for token attention) and not on $|V|$ or $|E|$. In this regime, the downstream computation is entirely instance-local: each prediction is a function of a single rooted neighborhood

and a small number of learned prompt parameters. From the standpoint of deployment, this implies that the resource bottleneck is not the global graph size, but rather local expansion (degree and r-hop growth) and the cost of embedding computation.

The accompanying lower bounds sharpen this statement by showing that the dependence on embedding dimension is not an artifact of the analysis. Even with unlimited computation and full knowledge of G, one cannot in general beat $K = \Omega(D/\varepsilon^2)$ for the class of bounded-norm linear predictors on D-dimensional frozen features. Thus, whenever we commit to a particular frozen representation and restrict downstream adaptation to a small head, the label complexity is governed by the statistical difficulty of linear prediction in that feature space, not by graph size. In other words, pre-training buys us something only insofar as it reduces the effective dimension of the task (e.g., by concentrating relevant variation into a low-dimensional subspace on which a small prompt can fit), or increases separability at bounded norm.

On the negative side, local access imposes an unavoidable barrier for tasks whose labels are not determined by bounded-radius neighborhoods. Our oracle lower bounds formalize a phenomenon frequently observed but rarely stated as a theorem: for global graph properties, or for prediction rules encoding global bits that are locally indistinguishable, any algorithm constrained to $q = o(|V|)$ neighborhood probes must fail with constant probability on worst-case graph families. This remains true regardless of how powerful the frozen encoder is, because the limitation is not representational but informational: the algorithm never observes enough of the graph to identify the target. The indistinguishability statement makes the same point in a sharper form: if two graph instances induce the same distribution of rooted r-neighborhoods at the queried points, then any local-access method has identical output distributions and hence cannot separate the instances beyond the Bayes error under that induced neighborhood distribution. Accordingly, the appropriate question for any downstream benchmark under local access is whether the label function is identifiable from the neighborhood oracle transcript given the permitted query budget.

These results suggest a concrete interpretation of "foundation" behavior for graph encoders under access constraints. A pre-trained (τ_ψ, f_θ) is useful downstream if it implements a map from local neighborhoods to embeddings that (a) retains the label-relevant r-local information while (b) rendering it easy for a small-P prompt to extract. In this view, the central scientific unknown is an *approximation* statement: for which families of graphs and which distributions over tasks does there exist a radius r and a frozen local encoder such that the induced feature map supports low-norm linear (or otherwise low-complexity) predictors? When such an approximation holds, the downstream learning problem reduces to classical generalization in a fixed feature space, and the oracle merely supplies those features. When it

fails, increasing P cannot compensate for missing oracle information, and increasing r or auxiliary retrieval must be treated as an explicit enlargement of the access budget rather than as "better prompting."

From a methodological standpoint, our framework also implies that meaningful evaluations of graph foundation models should report not only accuracy, but also the locality/access regime: radius r, the number of oracle calls, statistics of neighborhood size (e.g., $|E_r|$), and the adaptation budget P and K. Absent these quantities, improvements may conflate additional information (larger neighborhoods or more probes) with better representations. Moreover, the lower bounds recommend stress tests that deliberately vary locality: tasks that are r-local for small r should be solvable with few labels; tasks whose labels depend on slowly mixing or global structure should exhibit performance cliffs as r and the oracle budget are constrained.

Several open directions follow. First, one may seek distributional conditions under which the worst-case global-task lower bounds are avoidable, e.g., when graphs come from restricted generative models where global properties concentrate in local statistics. Second, one may formalize representation quality via neighborhood-kernel approximation or mutual information bounds between labels and embeddings conditioned on $O_G(v, r)$. Third, one may extend the analysis to settings in which local access is further restricted by privacy or rate limits, making the oracle transcript itself a constrained object. We regard these as natural next steps toward a theory in which the utility of graph foundation models is stated with the same explicitness as their access costs.