Yes, That's Mine: Asymptotically Foolproof LLM Ownership Identification Against Hidden Adversarial Decoding Parameter Perturbations

Author: Jerry Bao

Abstract
Creating and training an LLM can take vast amounts of resources, money, and effort. Because of this, many LLM creators seek to protect their credit and IP from being stolen. 'LLM ownership identification', the study of determining whether an unnamed LLM is a stolen version of a known LLM, is therefore one of the primary commercial motivators for the research of LLM provenance. White-box methods and backdoor-style black-box methods for LLM ownership identification face challenges with constrained practical assumptions and invasiveness, driving the search for effective and non-invasive black-box methods. There is a practice-theory gap in the current research for non-invasive, black-box methods, and we have proposed a community-wide formal program for classifying threat models which admit 'theoretically foolproof' (in the sense of decidable, statistical consistent, etc.) methods of LLM ownership identification. We argued that hidden decoding parameter perturbations are the necessary first model-identity-obfuscation class to investigate for this formal program, then presented statistically consistent tests for identifying a black-box LLM's ownership identity under such identity obfuscations. The methods presented in this paper satisfy key desirable properties such as: 1) being non-invasive and fully black box, 2) being statistically consistent, 3) being viable against an adversary with perfect information, 4) emitting useful intermediate results that may be useful in other research, and 5) being viable for further optimization. Along the way, we implicitly proved a fundamental negative result: standard, token-distribution-modifying decoding parameters are not obfuscating enough to prevent the existence of a non-invasive, theoretically foolproof method for determining a black-box LLM's ownership identity. In doing so, we established the first evidence of the feasibility of the proposed formal program, which aims to prove similar results under other threat scenarios.

Introduction:
Creating and training an LLM can take vast amounts of resources, money, and effort. Because of this, many LLM creators seek to protect their credit and IP from being stolen. 'LLM ownership identification', the study of determining whether an unnamed LLM is a stolen version of a known LLM, is therefore one of the primary commercial motivators for the research of LLM provenance. It sits alongside other related issues such as the detection of

malicious model tampering or the detection of worse, distilled, or quantized versions of an advertised model. With LLMs becoming increasingly proliferated and integrated into society, there is ever-increasing attention in researching methods to answering the question: "Is suspect model LLM B really a version of the original model LLM A?".

We briefly survey some of the most common methods in the literature for LLM ownership identification in the *stolen weights* threat scenario, where the goal is figuring out whether LLM B has stolen the weights of LLM A. We will also discuss some current challenges each approach faces, motivating the scope of this paper.

#1: White-box Methods

These methods often involve inspecting the parameters of LLM A and LLM B, deriving a signal from them, and comparing their similarity (https://arxiv.org/pdf/2312.04828, https://arxiv.org/pdf/2410.14273, https://arxiv.org/pdf/2511.06390, https://arxiv.org/pdf/2507.03014). The main limitation of this approach is that they assume white-box access to suspect model LLM B, which is often unrealistic in various practical scenarios, such as when an adversary deliberately hides LLM B's weights or when LLM B is not a stolen model, but its weights are still hidden due commercialization and IP protection. We will be focusing instead on black-box methods, where it is assumed that the weights of LLM B are not open to inspection.

#2: Backdoor-Style Black-Box Methods

Backdoor-style methods are among the most common black-box ownership identification approaches in the literature (https://arxiv.org/pdf/2401.12255, https://arxiv.org/pdf/2410.12318, https://arxiv.org/pdf/2509.03058, https://arxiv.org/pdf/2407.10887, https://ojs.aaai.org/index.php/AAAI/article/view/26750, https://arxiv.org/pdf/2402.14883, https://arxiv.org/pdf/2509.09703). They involve training the LLM to behave in an identifiable way (the observable) when a secret input (the trigger) is present in its prompt. Typically, the trigger is a string and the observable is an emitted response, but this is not a universal rule (see https://arxiv.org/pdf/2509.09703 and https://arxiv.org/pdf/2509.03058).

Backdoor-style methods are considered to be among the most reliable black-box approaches for LLM ownership identification. However, they are not without challenges that may dissuade usage. We will discuss a few practical, theoretical, and sociological challenges that this approach faces in the current day.

A key concern is that of *harmlessness*, the notion that backdoors should not interfere with general model utility. Backdoor-style methods necessarily require altering model weights and therefore model behavior. Such methods are known as 'invasive' methods. Poorly implemented backdoors risk accidental activation, normal task interference, and general

performance degradation. Backdoor method researchers are well-aware of this risk and attempt to maintain model utility, as well as conduct 'Harmlessness' tests to verify that the embedded backdoor does not impact normal task performance. However, these tests are primarily empirically validated, and even when they show no clear degradation in performance, face reproducibility and generalizability challenges.

As an illustrative example of practical reproducibility challenges, Instructional Fingerprinting (https://arxiv.org/pdf/2401.12255) reported that their fingerprinting causes no harm in downstream performance, based on tests involving several models evaluated on several benchmarks. A more recent paper (https://arxiv.org/pdf/2509.09703) reported that Instructional Fingerprinting caused notable performance degradations in their tests.

As for generalizability challenges, we note that the majority of papers on backdoor-style methods use small to mid-size LLMs. The generalization of harmlessness findings to large models is not thoroughly studied. For example, The Chain and Hash paper (https://arxiv.org/pdf/2407.10887) observes that the TruthfulQA benchmark (https://arxiv.org/pdf/2109.07958) demonstrates a significant improvement after fingerprint embedding compared to other benchmarks. The authors attribute these observed utility gains to the use of diverse prompt formatting templates during the embedding process. It's unclear whether these gains mask potential losses and whether they generalize to large, optimized models that have seen many formats in their training already.

Furthermore, we note that harmlessness tests in the literature primarily focus on performance-based tasks. The interaction and potential interference of these backdoor-style methods with other model objectives such as safety, alignment, and privacy are rarely considered. Systems engineering (Engineering a Safer World: Systems Thinking Applied to Safety) and sociological (Normal Accidents: Living with High-Risk Technologies) paradigms warn against increased operational complexity, noting that complex, not-well-understood interactions of components increase failure risks. We briefly describe some understudied risks here.

Taxonomic studies of jailbreaking (https://arxiv.org/pdf/2308.03825) and attempts to create universal jailbreaks for LLMs (https://arxiv.org/pdf/2405.20773) reveal a trend: jailbreaks almost always exploit a contextually-dependent policy that the LLM is intentionally trained on and which is beneficial under normal circumstances (e.g. roleplay framing, developer privileges, etc.). This suggests that introducing contextually-dependent policies switches into models could introduce more potentially exploitable vectors for misalignment, even without any malicious intent to do so. This is of particular concern to backdoors, which inherently train contextually-dependent policies into the model. Recent work (https://arxiv.org/pdf/2511.12414) evidences that seemingly benign backdoors can be used as a jailbreak to trigger misaligned behaviors without including any instructions to misalign during the fine-tune. Furthermore, other studies (https://arxiv.org/pdf/2508.14031,

https://arxiv.org/pdf/2310.03693, https://arxiv.org/pdf/2502.17424) have noted unintentional global alignment degradation of models from fine-tunes on seemingly neutral training data. Backdoors often require models to conceal secret behavior from users. The risks of training 'benign deception' into models on downstream model alignment are not well-studied in the context of backdoor-style methods for ownership identification.

Another potential interference that is understudied in this context is how the backdoors may interact with privacy mechanisms. For example, differential privacy (https://arxiv.org/pdf/1607.00133) is designed to limit the influence of any single training example on the learned model, which tends to reduce certain forms of memorization. Backdoor methods, on the other hand, rely on memorization-like phenomena, suggesting potential risks of interference if not carefully implemented. Studies like (https://arxiv.org/pdf/2311.06227, https://arxiv.org/pdf/2411.15831, https://arxiv.org/pdf/2504.21036) demonstrate that privacy mechanisms can interfere with backdoors in certain situations.

The importance of cautious investigation of the harmlessness of backdoor-style approaches stems from the fact that they are, fundamentally, invasive methods which alter underlying model weights.

#3: Non-Invasive Black-Box Methods
Ideally, a black-box method of LLM ownership identification is both reliable and non-invasive. This particular paper lies within the subdomain of *non-invasive, black-box* methods for LLM ownership identification.

There has been a significant body of research in non-invasive, black-box methods for LLM ownership identification. We identify 2 broad flavors of methods: 1) algorithmic/statistical (https://aclanthology.org/2025.findings-acl.546/, https://arxiv.org/pdf/2510.06605, https://arxiv.org/pdf/2505.12682, https://arxiv.org/pdf/2502.00706, https://arxiv.org/pdf/2405.02466) and 2) stylistic/behavioral (https://arxiv.org/pdf/2503.01659, https://arxiv.org/pdf/2408.02871, https://aclanthology.org/2021.findings-acl.409.pdf).

However, we identify a clear 'practice-theory gap', where we have many empirically effective methods but have an incomplete theoretical understanding of why they are so effective. This is not unusual among ML methods, as noted by survey papers such as https://arxiv.org/pdf/1807.03341 and among reactions in the ML community (https://www.science.org/content/article/ai-researchers-allege-machine-learning-alchemy). We observe that the vast majority of non-invasive, black-box methods for LLM ownership identification do not supply formal guarantees of effectiveness in the threat scenarios that they assume.

A Formal Program

We attempt to ameliorate the practice-theory gap by proposing a formal program wherein theoreticians attempt to classify the threat scenarios for non-invasive, black box model ownership identification that admit a 'theoretically foolproof' (in the sense of decidable, statistical consistent, etc.) guarantee of ownership identification. To evaluate the preliminary feasibility of such a program, we will attempt to show that such a guarantee exists for a well-motivated, nontrivial threat scenario.

Suppose an adversary were to steal a model's weights. Various perturbations to the deployed model, such as fine-tuning, decoding parameter alterations, system prompt alterations, etc. contribute to obfuscating the model's identity-attributable behavior during black-box tests of provenance. What should the first type of perturbation class that we investigate be? We argue that we should start by studying a *decoding-parameter-only* adversary. We argue this from two lenses: necessity and practicality.

Necessity
We identify decoding parameters as being uniquely the only *strictly necessary* part of an LLM's deployment stack other than the LLM's weights that modifies the LLM's statistical behavior. System prompts, safety filters, etc. are not strictly necessary for an LLM to be functional. Indeed, any working LLM must pass the token logit vector through a decoder in order to generate tokens. Thus, any formal analysis of a black-box LLM ownership identification method that claims to admit a theoretically foolproof guarantee must, at some point, confront decoding parameters head-on.

Practicality
Decoding parameter perturbations represent one of the most accessible, nontrivial perturbations that obfuscate an LLM's identity-attributable behavior. It is thus one of the most minimal, realistic perturbations that challenge identifiability. Decoding-parameter-only perturbations therefore serve as an opening challenge to the proposed formal program; if the program is to be viable and worthwhile to undertake, we should first demonstrate the ability to withstand this baseline threat scenario.

The scope of this paper
We will investigate whether there exists a theoretically foolproof, non-invasive, black-box method for identifying an LLM's ownership identity if we assume that the only allowable model perturbations are hidden perturbations to standard, token-distribution-modifying decoding parameters (defined later in this paper).

We will find that such a concrete method exists. It will furthermore satisfy the following desirable attributes:

1. Non-invasive and fully black box
2. Statistically consistent
3. Works even when the adversary has perfect information
4. Emits useful intermediate theoretical results that may potentially be useful in other research
5. Viable for further optimization

Along the way, we will have proven a fundamental negative result: standard, token-distribution-modifying decoding parameters are not obfuscating enough to prevent the existence of a non-invasive, theoretically foolproof method for determining a black-box LLM's ownership identity. Furthermore, we will have established the first evidence of the feasibility of the proposed formal program, which aims to prove similar results under other threat scenarios.

Setting:

Threat Model: LLM Ownership Identity Obfuscation via Standard Token-Distribution-Modifying Decoding Parameters
The following formal threat model provides the motivation for the rest of this paper's theoretical analysis:
- You are the owner of non-reasoning LLM A, which uses a random-sampling-based next-token decoder. You suspect that black-box non-reasoning LLM B, which also uses a random-sampling-based next-token decoder, is a copied instance of LLM A. You do not have access to LLM B's internals (e.g. weights, decoding parameters, etc.). You may call LLM B on whatever prompts you like, however many times as you like.
- Case 1: LLM B is a copy of LLM A. If LLM B is run by an adversary, the adversary will wish to obfuscate LLM B's provenance and the fact that it is a copy of LLM A by fixing a set of standard token-distribution-modifying decoding parameters one time and concealing their values. We assume the adversary has *perfect information,* and may fix the standard, token-distribution-modifying decoding parameters of LLM B according to this information.
- Case 2: LLM B is not a copy of LLM A. The actor behind LLM B does not have an adversarial incentive to impersonate LLM A.
- Excluded case: LLM B is derived from LLM A via a method other than modifying standard token-distribution-modifying decoding parameters.
- Goal: Determine whether we are in case 1 or 2. I.e. Determine whether or not LLM B is an instance of LLM A "up to decoding parameters".

Justification of assumptions & Threats not considered in this setting
Threat vectors not considered in our formal setting include minor model perturbations such

as fine-tuning and other methods for deriving models from LLM A other than perturbing standard token-distribution-modifying decoding parameters. The threat model used for our formal analysis explicitly excludes this possibility. Although we exclude fine-tuned or otherwise derived models from the scope of our analysis, we note that the arguments in this paper do not rely on the *full* strength of this assumption. We suggest a strong possibility that our methods may extend to certain carefully-defined subclasses of such models and encourage future work to attempt to cleanly classify practical subclasses of the excluded case for which the arguments and methods in this paper remain theoretically valid.

In addition, we do not consider the case where an adversary may deliberately attempt to have LLM B, when it is not a copy of LLM A, mimic the identity of LLM A. In our threat model, only copied models seek to conceal their identity; models not derived from LLM A are assumed not to engage in adversarial behavioral impersonation. Indeed, positive detection in our setting results in negative consequences for the actor behind LLM B, implying that a rational actor behind LLM B is disincentivized from impersonating LLM A (and it is furthermore in their interests to avoid being flagged as LLM A). This reflects the setting of IP protection rather than tamper or adversarial spoofing detection.

Although not formally analyzed here, we conjecture that, with respect to the identification methods developed in this work, it is extremely hard or infeasible for a model that is not LLM A to reliably impersonate LLM A. A formal treatment of this type of adversary is an interesting direction for future research. To avoid overclaiming, in this paper we restrict mathematical attention to our stated threat model.

Key Assumptions of the Threat Model:
Assumption 1: LLM B is assumed to be a non-reasoning LLM whose *only* implementation modifications from the base model are hidden, standard, fixed token-distribution-modifying decoding parameters. In particular, this means that LLM B is not allowed to use non-parametric modifications such as secret system prompts.

Assumption 2: "Standard token-distribution-modifying decoding parameters" is loosely defined as decoding parameters that modify the token distribution without resulting in any deterministic token generations. We list the allowed decoding parameters here:
- Temperature
- top-p (nucleus sampling)
- frequency penalty
- presence penalty
- repetition penalty
- logit_bias/token_bias: Let t be a token. Given a token logit z_t, then a bias b_t applied to the token transforms z_t to $z_t + b_t$.

- response_format / json_mode: A constrained-decoding mode that masks all tokens violating a JSON grammar, forcing the output to be valid JSON without modifying remaining logits.
- tool/function calling toggles: When toggled off, for each token in the tool-calling vocabulary, it transforms its logit z into $z - \lambda$, where $\lambda > 0$ is some constant that can be ∞ and is independent of the token.
- top_k
- typical_p: compute each token's self-information, sort tokens by increasing distance to the distribution's entropy, and keep the smallest prefix of tokens whose probabilities sum to $\geqslant$ typical_p
- no_repeat_ngram_size: disallow repeating any n-gram of size no_repeat_ngram_size in the output
- Mirostat
- Mirostat 2
- Top-a/η-sampling: Truncates tokens that have p values such that $p < p_{max}^{\eta}$
- min-p: A token logit masking rule
- tail-free sampling: A token logit masking rule

Definitions
- LLM Equality: We say LLM A = LLM B, LLM B is a copy of LLM A, or LLM B is identical to LLM A if LLM A and LLM B share identical architecture, weights, vocabulary, tokenizers, and all other implementation details that would constitute a base LLM model. Note that the values of auxiliary parameters such as temperature or logit bias are not included among the implementation details used for establishing LLM equality.
- If we have distinct contexts/prompts $C_1,..., C_m$, we say that the distinct strings $a_1,..., a_k, b_1,..., b_k$ satisfy the (†) Regularity Conditions with respect to $C_1,..., C_m$ if: 1) each of $a_1,..., a_k, b_1,..., b_k$ is a token in the vocabulary of LLM A, 2) given any generated output string from LLM A, $t \in \{a_1,..., a_k, b_1,..., b_k\}$ is a prefix of that string iff it is the first generated token of that string (practical methods for ensuring this are discussed in the Appendix), 3) none of $a_1,..., a_k, b_1,..., b_k$ appears as a token within $C_j, j = 1 : m$ with respect to LLM A's own tokenization (practical case: none of $a_1,..., a_k, b_1,..., b_k$ is a substring of $C_j, j = 1 : m$), 4) for each a_i and each C_j, it is *possible* for LLM B (after fixing its hidden decoding parameters) to generate an output string on context C_j with a_i as that output's prefix string, and 5) for each b_i and each C_j, it is *possible* for LLM B (after fixing its hidden decoding parameters) to generate an output string on context C_j with b_i as that output's prefix string.
- Suppose we repeatedly call LLM B on each context C_j. Let N_j be the number of times LLM B was called on context C_j (determined by us)
- Let T be LLM B's temperature (unknown to us)

- For LLM A, let $z_{a_i,j}$ be token a_i's base logit given context C_j. (known to us)
- For LLM A, Let $z_{b_i,j}$ be token b_i's base logit given context C_j. (known to us)
- Let $z_{a,j} = \sum_{i=1}^{k} z_{a_i,j}$
- Let $z_{b,j} = \sum_{i=1}^{k} z_{b_i,j}$
- Let $\Delta z_j = z_{a,j} - z_{b,j}$
- Let $p_j(a, i)$ be the true probability of a_i being a prefix string of a generated output from LLM B on context C_j (unknown to us)
- Let $p_j(b, i)$ be the true probability of b_i being a prefix string of a generated output from LLM B on context C_j (unknown to us)
- Let $n_j(a, i)$ be the actual number of times a_i appeared as a prefix string of a generated output from LLM B on context C_j (known to us)
- Let $n_j(b, i)$ be the actual number of times b_i appeared as a prefix string of a generated output from LLM B on context C_j (known to us)
- Let $g_j = ln\left(\prod_{i=1}^{k} \frac{n_j(a, i)}{n_j(b, i)} \right) = \sum_{i=1}^{k} ln(n_j(a, i)) - \sum_{i=1}^{k} ln(n_j(b, i))$
- Let $c_j = ln\left(\prod_{i=1}^{k} \frac{p_j(a, i)}{p_j(b, i)} \right) = \sum_{i=1}^{k} ln(p_j(a, i)) - \sum_{i=1}^{k} ln(p_j(b, i))$

Comment: The † regularity conditions are not intended to be an 'uncontrollable assumption'. They are intended to be *by construction.* The choices of $C_1,..., C_m$ and $a_1,..., a_k, b_1,..., b_k$ are within the user's control.

We immediately have the following theorem:

Theorem 1: g_j is a consistent estimator of c_j.

Proof: Follows from the fact that $\dfrac{n_j(a, i)}{n_j(b, i)} = \dfrac{n_j(a, i) / N_j}{n_j(b, i) / N_j}$ are consistent estimators of $\dfrac{p_j(a, i)}{p_j(b, i)}$.

$\square$

Theoretical Consequences of LLM A = LLM B

Let C₁,C₂ be 2 prompts. Let a₁,...,aₖ,b₁,...,bₖ be 2k strings. For the rest of this section, we shall assume that a₁,...,aₖ,b₁,...,bₖ satisfy † with respect to C₁,C₂. Furthermore, for the rest of this section, we shall assume LLM A = LLM B.

Note that by $†, 1$ then $a_1,..., a_k, b_1,..., b_k$ simply becomes a list of tokens of LLM B. Furthermore, under $†, 2$, some of our previous definitions equivalently become:

- Let $p_j(a, i)$ be the true probability of a_i in the first-token distribution of LLM B with respect to context C_j (unknown to us)
- Let $p_j(b, i)$ be the true probability of b_i in the first-token distribution of LLM B with respect to context C_j (unknown to us)
- Let $n_j(a, i)$ be the actual number of times a_i appeared as the first token of LLM B on context C_j (known to us)
- Let $n_j(b, i)$ be the actual number of times b_i appeared as the first token of LLM B on context C_j (known to us)

We then provide some additional definitions:

- Let b_{a_i} be token a_i's logit bias in LLM B (unknown to us)
- Let b_{b_i} be token b_i's logit bias in LLM B (unknown to us)
- Let $b_a = \sum_{i=1}^{k} b_{a_i}$
- Let $b_b = \sum_{i=1}^{k} b_{b_i}$
- Let $\Delta b = b_a - b_b$

Theorem 2: Suppose LLM A = LLM B. Fix the context to be C_1 or C_2. For all i in $1 : k$, the value $\dfrac{p_j(a, i)}{p_j(b, i)}$ is invariant under all standard token-distribution-modifying decoding parameters applied to LLM B, except for temperature T and logit biases $b_{a_1},..., b_{a_k}, b_{b_1}..., b_{b_k}$.

Proof:
We first consider the following decoding parameters:

- top-p (nucleus sampling)
- response_format / json_mode: A constrained-decoding mode that masks all tokens violating a JSON grammar, forcing the output to be valid JSON without modifying remaining logits.
- top_k
- typical_p: compute each token's self-information, sort tokens by increasing distance to

the distribution's entropy, and keep the smallest prefix of tokens whose probabilities sum to $\geqslant$ typical_p
- no_repeat_ngram_size: disallow repeating any n-gram of size no_repeat_ngram_size in the output
- Mirostat
- Top-a/η-sampling: Truncates tokens that have p values such that $p < p_{max}^{\eta}$
- min-p: A token logit masking rule
- tail-free sampling: A token logit masking rule

These are all token masking decoding parameters. In other words, they each define a set S of allowed tokens. Note that a_i and b_i are assumed to be in S due to †, 4 and †, 5.

Before masking, $\dfrac{p_j(a, i)}{p_j(b, i)} = \dfrac{\frac{e^{z_{a_i}}}{\sum_t e^{z_t}}}{\frac{e^{z_{b_i}}}{\sum_t e^{z_t}}} = \dfrac{e^{z_{a_i}}}{e^{z_{b_i}}}$, where z is each token's logit. After masking,

$\dfrac{p_j(a, i)}{p_j(b, i)} = \dfrac{\frac{e^{z_{a_i}}}{\sum_{t \in S} e^{z_t}}}{\frac{e^{z_{b_i}}}{\sum_{t \in S} e^{z_t}}} = \dfrac{e^{z_{a_i}}}{e^{z_{b_i}}}$. We conclude that none of the masking decoding parameters affects

$\dfrac{p_j(a, i)}{p_j(b, i)}$.

We next consider the following 3 decoding parameters:
- frequency penalty
- presence penalty
- repetition penalty

Note that these 3 decoding parameters only kick in if a_i or b_i actually appear in the context. But by †, 3, neither of them appears in the context. Therefore, neither z_{a_i} or z_{b_i} is modified.

Using a similar argument as before, we see that $\dfrac{p_j(a, i)}{p_j(b, i)} = \dfrac{e^{z_{a_i}}}{e^{z_{b_i}}}$ both before and after each of these decoding parameters is applied. We conclude that none of these 3 decoding parameters affects $\dfrac{p_j(a, i)}{p_j(b, i)}$.

We next consider the following decoding parameter:
- tool/function calling toggles: When toggled off, for each token in the tool-calling vocabulary, it transforms its logit z into $z - \lambda$, where $\lambda > 0$ is some constant that can be ∞ and is independent of the token.

Note that if $\lambda = \infty$, then this is a masking decoding parameter. In this case, the same reasoning as before applies for why $\dfrac{p_j(a, i)}{p_j(b, i)}$ is unaffected. Else, this is just a standard token logit bias, which is permitted within the assumptions of theorem 2.

Finally, we consider the following decoding parameter.
  • Mirostat 2
Note that Mirostat 2 does not affect the logit values of surviving tokens in the first-token distribution. This is because Mirostat 2 needs adaptive feedback from earlier tokens in the output and none exist for the first token. We can therefore apply our reasoning from the masking case to conclude that $\dfrac{p_j(a, i)}{p_j(b, i)}$ is unaffected by Mirostat 2.

Finally, we can conclude that only temperature and logit bias affect the ratio $\dfrac{p_j(a, i)}{p_j(b, i)}$. This completes the proof of the claim. □

Corollary 1: Suppose LLM A = LLM B. Fix the context to be C_1 or C_2. The value $\displaystyle\prod_{i=1}^{k} \dfrac{p_j(a, i)}{p_j(b, i)}$ (and therefore c_j) with respect to the first-token distribution is invariant under all standard token-distribution-modifying decoding parameters, except for temperature and bias. □

Corollary 2: Suppose LLM A = LLM B. Then $c_j = \ln\left(\displaystyle\prod_{i=1}^{k} \dfrac{p_j(a, i)}{p_j(b, i)}\right) = \dfrac{\Delta z_j + \Delta b}{T}$

Proof:
Since only temperature and logit bias affect probability ratios (theorem 2), when computing $\dfrac{p_j(a, i)}{p_j(b, i)}$, we may assume only only temperature and bias affect the token logits. Thus:

$$\frac{p_j(a, i)}{p_j(b, i)} = \frac{exp\left(\frac{z_{a_i,j}+b_{a_i}}{T}\right) / \sum_t exp\left(\frac{z_{t,j}+b_t}{T}\right)}{exp\left(\frac{z_{b_i,j}+b_{b_i}}{T}\right) / \sum_t exp\left(\frac{z_{t,j}+b_t}{T}\right)}$$

$$= exp\left(\frac{(z_{a_i,j} - z_{b_i,j}) + (b_{a_i} - b_{b_i})}{T}\right)$$

Thus:

$$\prod_{i=1}^{k} \frac{p_j(a, i)}{p_j(b, i)} = \prod_{i=1}^{k} exp\left(\frac{(z_{a_i,j} - z_{b_i,j}) + (b_{a_i} - b_{b_i})}{T}\right)$$

$$= exp\left(\sum_{i=1}^{k} \frac{(z_{a_i,j} - z_{b_i,j}) + (b_{a_i} - b_{b_i})}{T}\right)$$

$$= exp\left(\frac{\Delta z_j + \Delta b}{T}\right)$$

Therefore, $c_j = ln\left(\prod_{i=1}^{k} \frac{p_j(a, i)}{p_j(b, i)}\right) = \frac{\Delta z_j + \Delta b}{T}.$ □

Temperature Estimation:

Theorem 3: Suppose LLM A = LLM B. Then $T = \dfrac{\Delta z_1 - \Delta z_2}{c_1 - c_2}$

Proof:

$$T = \frac{\Delta z_1 - \Delta z_2}{c_1 - c_2}$$

$$= \frac{\Delta z_1 - \Delta z_2}{\frac{\Delta z_1 + \Delta b}{T} - \frac{\Delta z_2 + \Delta b}{T}} \text{ (by corollary 2)}$$

$$= T \cdot \frac{\Delta z_1 - \Delta z_2}{\Delta z_1 - \Delta z_2} \text{ (1)}$$

$$= T$$ □

Theorem 4: Suppose LLM A = LLM B. Then $\dfrac{\Delta z_1 - \Delta z_2}{g_1 - g_2}$ is a consistent estimator of T.

Proof:

This follows from theorem 1 and theorem 3. □

Logit Bias Displacement Estimation

Theorem 5: Suppose LLM A = LLM B. Then $\Delta b = \dfrac{c_2 \Delta z_1 - c_1 \Delta z_2}{c_1 - c_2}$

Proof:

By Corollary 2, we have:

$$\frac{c_2 \Delta z_1 - c_1 \Delta z_2}{c_1 - c_2} = \frac{\frac{\Delta z_2 + \Delta b}{T} \Delta z_1 - \frac{\Delta z_1 + \Delta b}{T} \Delta z_2}{\frac{\Delta z_1 + \Delta b}{T} - \frac{\Delta z_2 + \Delta b}{T}}$$

$$= \frac{(\Delta z_2 + \Delta b)\Delta z_1 - (\Delta z_1 + \Delta b)\Delta z_2}{\Delta z_1 - \Delta z_2}$$

$$= \Delta b$$

$\square$

Theorem 6: Suppose LLM A = LLM B. Then $\dfrac{g_2 \Delta z_1 - g_1 \Delta z_2}{g_1 - g_2}$ is a consistent estimator of Δb.

Proof: This follows from theorem 1 and theorem 5. $\square$

Discussion:

Let us remind ourselves what Δb is: it is the difference between the sum of the logit biases of the a_i's and the sum of the logit biases of the b_i's. If $k = 1$, then Δb is simply the difference between the logit biases of a_1 and b_1. Let $a := a_1$ and $b := b_1$. In this case, we may write Δb as Δb_{a-b}. Introduce a 3rd token c. Using theorem 6, we can asymptotically compute $\Delta b_{a-b}, \Delta b_{b-c}, \Delta b_{c-a}$. This gives us a system of 3 linear equations where the variables are the logit biases b_a, b_b, and b_c. Solving gives us the logit biases of a, b, c.

Our discussion informs us that, assuming LLM A = LLM B, we can theoretically recover the individual logit biases that were applied to specific tokens in LLM B to an arbitrarily high degree of accuracy. This is perhaps a somewhat surprising identifiability result. Moreover, combining this with theorem 4 tells us that when LLM A = LLM B, logit biases and temperature are both recoverable information, even if we did not initially know which standard token-distribution-modifying decoding parameters LLM B used.

Theoretical Consequences of LLM A ≠ LLM B

Let C_1,C_2 be 2 prompts. Let a_1,...,a_k,b_1,...,b_k be 2k strings. For the rest of this section, we shall assume that a_1,...,a_k,b_1,...,b_k satisfy † with respect to C_1,C_2.

Definitions:

- Let $est(C_1; C_2; a_1,..., a_k; b_1,..., b_k) = \dfrac{\Delta z_1 - \Delta z_2}{g_1 - g_2}$

- Let $est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k) = \dfrac{g_2 \Delta z_1 - g_1 \Delta z_2}{g_1 - g_2}$

It is trivial to see that even though LLM A $\neq$ LLM B, est and $est2$ are still consistent estimators. Namely, we have:

Corollary 3: $est(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ is a consistent estimator of $\dfrac{\Delta z_1 - \Delta z_2}{c_1 - c_2}$. $\qquad \square$

Corollary 4: $est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ is a consistent estimator of $\dfrac{c_2 \Delta z_1 - c_1 \Delta z_2}{c_1 - c_2}$. $\qquad \square$

<u>Determining LLM Ownership Identity</u>

We now have the tools to construct specific tests for LLM ownership identity. We will first present the most general form of the framework, then illustrate specific examples afterwards.

But first, let us introduce some new definitions that generalize our old definitions:

Definitions:
- If we have contexts/prompts $C_1,..., C_m$, we say that the strings $a_{1,j},..., a_{k_j,j}, b_{1,j},..., b_{k_j,j}$ for $j = 1:m$ satisfy the (††) Regularity Conditions with respect to $C_1,..., C_m$ if: 1) for each $a_{i,j}$ and each C_j, it is *possible* for LLM B (after fixing its hidden decoding parameters) to generate an output string on context C_j with $a_{i,j}$ as that output's prefix string, and 2) for each $b_{i,j}$ and each C_j, it is *possible* for LLM B (after fixing its hidden decoding parameters) to generate an output string on context C_j with $b_{i,j}$ as that output's prefix string.
- Suppose we repeatedly call LLM B on each context C_j. Let N_j be the number of times LLM B was called on context C_j (determined by us).
- Let $p_j(a, i)$ be the true probability of $a_{i,j}$ being a prefix string of a generated output from LLM B on context C_j (unknown to us)
- Let $p_j(b, i)$ be the true probability of $b_{i,j}$ being a prefix string of a generated output from LLM B on context C_j (unknown to us)
- Let $n_j(a, i)$ be the actual number of times $a_{i,j}$ appeared as a prefix string of a generated output from LLM B on context C_j (known to us)
- Let $n_j(b, i)$ be the actual number of times $b_{i,j}$ appeared as a prefix string of a generated output from LLM B on context C_j (known to us)
- Let $g_j = ln\left(\prod\limits_{i=1}^{k_j} \dfrac{n_j(a, i)}{n_j(b, i)} \right) = \sum\limits_{i=1}^{k_j} ln(n_j(a, i)) - \sum\limits_{i=1}^{k_j} ln(n_j(b, i))$

- Let $c_j = ln\left(\prod_{i=1}^{k_j} \dfrac{p_j(a, i)}{p_j(b, i)}\right) = \sum_{i=1}^{k_j} ln(p_j(a, i)) - \sum_{i=1}^{k_j} ln(p_j(b, i))$

Comment: The †† regularity conditions are not intended to be an 'uncontrollable assumption'. They are intended to be *by construction.* The choices of $C_1,..., C_m$ and $a_{1,j},..., a_{k_j,j}, b_{1,j},..., b_{k_j,j}; j = 1 : m$ are within the user's control.

We must also introduce the concept of an *experiment.*

What is an experiment?
Note that each $n_j(a, i)$ comes from running an experiment that calls LLM B multiple times on context C_j and recording the number of times that $a_{i,j}$ appears as an output prefix. Similar story for $n_j(b, i)$. It must be clarified that the actual sequence of runs we make with LLM B on context C_j might be *shared* between different j's.

Consider the following illustrative scenario: Fix context C and let $C_{j_1} = C_{j_2} = C$. For C_{j_1}, associate the strings $a_{1,j_1},..., a_{k_{j_1},j_1}, b_{1,j_1},..., b_{k_{j_1},j_1}$. For C_{j_2}, associate the strings $a_{1,j_2},..., a_{k_{j_2},j_2}, b_{1,j_2},..., b_{k_{j_2},j_2}$, which may or may not overlap with $a_{1,j_1},..., a_{k_{j_1},j_1}, b_{1,j_1},..., b_{k_{j_1},j_1}$. Now run LLM B repeatedly on context $C = C_{j_1} = C_{j_2}$. This sequence of runs constitutes a single experiment. From this single experiment, we may record the number of output prefix occurrences of $a_{1,j_1},..., a_{k_{j_1},j_1}, b_{1,j_1},..., b_{k_{j_1},j_1}$ (i.e. record $n_{j_1}(a, i)$ and $n_{j_1}(b, i)$) and the number of output prefix occurrences of $a_{1,j_2},..., a_{k_{j_2},j_2}, b_{1,j_2},..., b_{k_{j_2},j_2}$ (i.e. record $n_{j_2}(a, i)$ and $n_{j_2}(b, i)$). Crucially, note that g_{j_1} and g_{j_2} are not built from independent, self-contained experiments. They are built from the *same* experiment and therefore coupled and non-independent.

The key point is this: different j's do not necessarily represent different experiments. Different j's may represent recorded information from the *same* experiment. Not merely from a *copy* of an experiment, but from the *same* experiment.

We can now define what an experiment is.

Definitions:
- An experiment E encapsulates all the information about the full sequence of runs we do with LLM B on a fixed context. Output prefix count data is recorded from this experiment and may be used for multiple different j's.
- Let $E_1,..., E_w$ be the list of experiments we have done.
- For each $r = 1 : w$, let O_r be the subset of $n_j(a, i)$ and $n_j(b, i)$ observations that came

from experiment E_r.

- Let $k^{(r)}$ be the number of elements in $O^{(r)}$

Here are some easy facts about O_r:

- The O_r's partition the $n_j(a, i)$'s and $n_j(b, i)$'s. After all, each $n_j(a, i)$ or $n_j(b, i)$ may only come from a single experiment E_r.
- The different O_r's are statistically independent of each other. After all, they record observations from independent experiments.
- For each j, $n_j(a, i)$ and $n_j(b, i)$ for $i = 1 : k_j$ all belong to the same O_r. This justifies the notation $j \in E_r$, which is taken to mean that the observations for j (all) came from E_r.
- Each O_r is associated with a number $N^{(r)}$ representing the number of times LLM B was run/called during the experiment. Note that for any $j = 1 : m$, if $j \in E_r$, then $N^{(r)} = N_j$.
- For any j_1, j_2, if $j_1 \in E_r$ and $j_2 \in E_r$, then $C_{j_1} = C_{j_2}$.
- For any $j = 1 : m$, $j \in E_r$ for some $r = 1 : w$.

We now have the language to talk about the general LLM ownership identification method.

The General Method

> Throughout this "The General Method" subsection, we will be assuming †† instead of † (which is a special case of ††; †† is a relaxation of †).

<u>The Estimator Algorithm</u>

1. For each j, construct estimator $g_j = ln \left(\prod_{i=1}^{k_j} \frac{n_j(a, i)}{n_j(b, i)} \right)$

2. Construct a real-analytic function $\phi(x_1, ..., x_m) : \mathbb{R}^m \to \mathbb{R}$ such that $\phi(g_1, ..., g_m)$ is a consistent estimator of a known constant C if LLM A = LLM B. $\phi(x_1, ..., x_m)$ must not be identically C.

3. Decision rule: If $\phi(g_1, ..., g_m)$ converges to C as $N^{(1)}, ..., N^{(w)} \to \infty$, conclude that LLM A = LLM B. Else, conclude that LLM A $\neq$ LLM B.

Properties of the estimator algorithm:

1. $\phi(g_1, ..., g_m)$ is a convergent statistic as $N^{(1)}, ..., N^{(w)} \to \infty$.
2. If LLM A = LLM B, this statistic converges to C. Else, it converges to a non-C value with probability 1.

We shall defer the proof of the estimator algorithm's properties to the end of this section (★).

We can also use this same $\phi(g_1,..., g_m)$ to construct a hypothesis test:

The Hypothesis Test
Let $\phi(g_1,..., g_m)$ be from step 2 of the estimator algorithm.
Pick any function $\epsilon\left(N^{(1)},..., N^{(w)}\right)$ such that $\epsilon \downarrow 0$ and $\dfrac{\epsilon}{\sqrt{\sum_{r=1}^{w} 1/N^{(r)}}} \to \infty$ as

$N^{(1)},..., N^{(w)} \to \infty$ (e.g. $\epsilon\left(N^{(1)},..., N^{(w)}\right) = min\left(N^{(1)},..., N^{(w)}\right)^{-\alpha}$ where $0 < \alpha < 1/2$)
Let H_0 be the null hypothesis that LLM A = LLM B.
Let H_1 be the alternate hypothesis that LLM A $\neq$ LLM B.
Assume $N^{(1)},..., N^{(w)} \to \infty$.

Decision rule: If $|\phi(g_1,..., g_m) - C| > \epsilon\left(N^{(1)},..., N^{(w)}\right)$, reject H_0. Else, fail to reject H_0.

Properties of the hypothesis test:
    1. If H_0 if true, with probability 1, the test does not reject H_0 as $N^{(1)},..., N^{(w)} \to \infty$.
    2. If H_1 is true, with probability 1, the test rejects H_0 as $N^{(1)},..., N^{(w)} \to \infty$.
Thus, the hypothesis test is consistent in the sense of properties 1 and 2. We shall defer the proof of the hypothesis test's properties to the end of this section (★).

Specific Examples
We will present 2 illustrative methods for determining LLM ownership identity using the general framework that was just presented. In particular, we will show that it is possible to construct the estimator $\phi(g_1,..., g_m)$ and corresponding known constant C which form the crux of the estimator algorithm and hypothesis test.

The first estimator will be built from est and the second estimator will be built from $est2$. Intuitively, the $\phi(g_1,..., g_m)$ estimator that we construct will exploit a sharp dichotomy between the cases of LLM A = LLM B and LLM A $\neq$ LLM B: conditional stability. When LLM A = LLM B, then est and $est2$ will always converge to fixed and stable quantities T and Δb. However, when LLM A $\neq$ LLM B, the limits of est and $est2$ will vary noisily and unpredictably, determined by how the specific instances of est and $est2$ were constructed.

Because the goal here is to be illustrative and to demonstrate the *existence* of such methods, do not expect the estimators constructed in this section to be the fastest-converging estimators that can be constructed from est and $est2$.

<u>Naive Method I</u>
Step 1:
Take contexts C_1, C_2 and strings $a_1,..., a_k$ and $b_1,..., b_k$ satisfying the (†) conditions. Similarly, take contexts C_3, C_4 and strings $a_1',..., a_k'$ and $b_1',..., b_k'$ satisfying the (†) conditions.

Step 2:
Relabel to the notation of this section:
- Let $\langle a_{1,1},..., a_{k_1,1}, b_{1,1},..., b_{k_1,1} \rangle = \langle a_{1,2},..., a_{k_2,2}, b_{1,2},..., b_{k_2,2} \rangle := \langle a_1,..., a_k, b_1,..., b_k \rangle$
- Let $\langle a_{1,3},..., a_{k_3,3}, b_{1,3},..., b_{k_3,3} \rangle = \langle a_{1,4},..., a_{k_4,4}, b_{1,4},..., b_{k_4,4} \rangle := \langle a_1',..., a_k', b_1',..., b_k' \rangle$

Step 3:

For each $j = 1:4$, construct estimator $g_j = ln\left(\prod_{i=1}^{k_j} \dfrac{n_j(a, i)}{n_j(b, i)} \right)$ using C_j and

$a_{1,j},..., a_{k_j,j}, b_{1,j},..., b_{k_j,j}$.

Step 4:
Define:
- For LLM A, let $z_{a_{i,j}}$ be token $a_{i,j}$'s base logit given context C_j. (known to us)
- For LLM A, Let $z_{b_{i,j}}$ be token $b_{i,j}$'s base logit given context C_j. (known to us)
- Let $z_{a,j} = \sum_{i=1}^{k} z_{a_{i,j}}$
- Let $z_{b,j} = \sum_{i=1}^{k} z_{b_{i,j}}$
- Let $\Delta z_j = z_{a,j} - z_{b,j}$

Choose $\phi(x_1, x_2, x_3, x_4) = \dfrac{\Delta z_1 - \Delta z_2}{x_1 - x_2} - \dfrac{\Delta z_3 - \Delta z_4}{x_3 - x_4}$.

Step 5:
We have
$\phi(g_1, g_2, g_3, g_4) = est(C_1; C_2; a_1,..., a_k; b_1,..., b_k) - est(C_3; C_4; a_1',..., a_k'; b_1',..., b_k')$.
Let $C = 0$.

Claim: $\phi(g_1, g_2, g_3, g_4)$ is a consistent estimator of $C = 0$ if LLM A = LLM B.

Proof:
Assume LLM A = LLM B.

By theorem 4, both $est(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ and $est(C_3; C_4; a_1',..., a_{k'}'; b_1',..., b_{k'}')$ converge to T, so $\phi(g_1, g_2, g_3, g_4)$ converges to 0. $\qquad\qquad$ □

Naive Method II

Step 1:

Take contexts C_1, C_2, C_3, C_4 and strings $a_1,..., a_k$ and $b_1,..., b_k$ satisfying the † regularity conditions for both (C_1, C_2) and (C_3, C_4).

Step 2:

Relabel to the notation of this section:

$\langle a_{1,j},..., a_{k_j,j}, b_{1,j},..., b_{k_j,j} \rangle := \langle a_1,..., a_k, b_1,..., b_k \rangle$ for $j = 1:4$.

Step 3:

For each $j = 1:4$, construct estimator $g_j = ln\left(\prod_{i=1}^{k_j} \dfrac{n_j(a, i)}{n_j(b, i)} \right)$ using C_j and

$a_{1,j},..., a_{k_j,j}, b_{1,j},..., b_{k_j,j}$.

Step 4:

Define:

- For LLM A, let $z_{a_{i,j}}$ be token $a_{i,j}$'s base logit given context C_j. (known to us)
- For LLM A, Let $z_{b_{i,j}}$ be token $b_{i,j}$'s base logit given context C_j. (known to us)
- Let $z_{a,j} = \sum_{i=1}^{k} z_{a_{i,j}}$
- Let $z_{b,j} = \sum_{i=1}^{k} z_{b_{i,j}}$
- Let $\Delta z_j = z_{a,j} - z_{b,j}$

Choose $\phi(x_1, x_2, x_3, x_4) = \dfrac{x_2 \Delta z_1 - x_1 \Delta z_2}{x_1 - x_2} - \dfrac{x_4 \Delta z_3 - x_3 \Delta z_4}{x_3 - x_4}$ and let $C = 0$.

Step 5:

We have

$\phi(g_1, g_2, g_3, g_4) = est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k) - est2(C_3; C_4; a_1,..., a_k; b_1,..., b_k)$.
Let $C = 0$.

Claim: $\phi(g_1, g_2, g_3, g_4)$ is a consistent estimator of $C = 0$ if LLM A = LLM B.

Proof:

Assume LLM A = LLM B.

By theorem 6, both $est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ and $est2(C_3; C_4; a_1,..., a_k; b_1,..., b_k)$ converge to Δb, so $\phi(g_1, g_2, g_3, g_4)$ converges to 0. □

Deferred Proofs (★)

Lemma 1: g_j is a consistent estimator of c_j for $j = 1 : m$ as $N^{(1)},..., N^{(w)} \to \infty$.

Proof: As $N^{(1)},..., N^{(w)} \to \infty$, we have $N_1,..., N_m \to \infty$. The lemma then follows from the fact that $\dfrac{n_j(a, i)}{n_j(b, i)} = \dfrac{n_j(a, i) / N_j}{n_j(b, i) / N_j}$ are consistent estimators of $\dfrac{p_j(a, i)}{p_j(b, i)}$. □

Lemma 2: $\phi(g_1,..., g_m)$ is a consistent estimator of $\phi(c_1,..., c_m)$ as $N^{(1)},..., N^{(w)} \to \infty$.
Proof: Follows from lemma 1. □

Lemma 3: The solution set $S = \left\{ x \in \mathbb{R}^m : \phi(x) = C \right\}$ has Lebesgue measure 0.

Proof: $\phi(x)$ (and therefore $\phi(x) - C$) is real-analytic by assumption. Furthermore, $\phi(x) - C$ is not identically zero by assumption. A standard result in real-analytic geometry states that the zero set of a nontrivial real-analytic function on $\mathbb{R}^m$ is a countable union of manifolds of dimension at most $m - 1$, and therefore has Lebesgue measure zero. □

Now consider $c = (c_1,..., c_m) = \left(ln\left(\displaystyle\prod_{i=1}^{k_1} \frac{p_1(a, i)}{p_1(b, i)} \right),..., ln\left(\displaystyle\prod_{i=1}^{k_m} \frac{p_m(a, i)}{p_m(b, i)} \right) \right)$. Let Θ denote the configuration of LLM B, including architecture, model weights, and decoding parameters. Let $c(\Theta)$ (or just c when the context is clear) denote the values of $(c_1,..., c_m)$ corresponding to Θ. When LLM A $\neq$ LLM B, we treat $c(\Theta)$ as a random vector over the distribution of "potential" LLM configurations Θ.

The rest of our proofs require the following axiom:

Axiom 1: If LLM A $\neq$ LLM B, then $c(\Theta) \sim \pi$ where π is a probability measure with $\pi(S) = 0$.

Interpretive modeling justification of Axiom 1:
Before we continue, note that nothing below is meant to prove the axiom (as indeed, the axiom is logically independent of our previous assumptions). This is a modeling narrative aiming to justify the axiom as a principled modeling assumption (analogous to the role the Church-Turing Thesis plays in computability theory, which links a formal mathematical

condition to an informal notion of real-world behavior and is justified by modeling considerations rather than proof).

Lemma 3 demonstrates that S is a 'pathological' set in the sense of having measure 0, being so small that it takes up 0 measurable space within $\mathbb{R}^m$. LLM A = LLM B can be viewed as a pathological special case that collapses c onto this low-dimensional subset S.

On the other hand, when LLM A $\neq$ LLM B, we are in the 'generic case' where there is no reason, so to speak, for c to collapse onto S. In fact, one might go a step further to argue that c should be viewed as an absolutely continuous random vector in $\mathbb{R}^m$. Indeed, if assumed, this would imply Axiom 1 as a corollary since any absolutely continuous probability measure with respect to Lebesgue measure necessarily assigns probability 0 to S. However, we choose to keep Axiom 1 as our most minimal assumption. It claims that if LLM A $\neq$ LLM B, then the probability of c landing on the pathological set S is 0.

Implicitly, we've been assuming that when LLM A $\neq$ LLM B, LLM B's configuration is drawn from a 'natural' and 'non-adversarial' distribution of potential LLM configurations. This is not the case if the actor behind LLM B artificially manipulates LLM B's configuration in such a precise and specific way to force c to land on S. We argue that we should assume that this doesn't happen under our assumed threat model. Why should we assume that the actor behind LLM B doesn't attempt to force $c \in S$ to happen when LLM A $\neq$ LLM B? This question concerns the incentives of a rational actor behind LLM B given our threat model. We assert the actor has a disincentive to deliberately configure LLM B in such a way such that $c \in S$ happens.

Suppose LLM A $\neq$ LLM B. Note that if $\phi(g_1,...,g_m)$ converges to C, both the estimator algorithm and the hypothesis test will erroneously detect LLM B as being LLM A as $N^{(1)},..., N^{(w)} \to \infty$. This isn't obvious for the case of the hypothesis test. The later proof for property 1 of the hypothesis test will show that if $\phi(g_1,...,g_m)$ converges to C, then the hypothesis test will decide LLM A = LLM B (i.e. fail to reject H_0) as $N^{(1)},..., N^{(w)} \to \infty$. Crucially, the proof of property 1 of the hypothesis test will not require assuming Axiom 1 (or else we would have circular reasoning). By lemma 2, $\phi(g_1,...,g_m)$ being a consistent estimator of C is equivalent to $\phi(c_1,...,c_m) = C$. By definition of S, $\phi(c_1,...,c_m) = C$ is equivalent to $c \in S$. Therefore, satisfying the condition $c \in S$ causes both the estimator algorithm and hypothesis test to erroneously classify LLM B as being identical to LLM A. Thus, the motive of an actor attempting to configure LLM B in such a way as to force $c \in S$ is to impersonate LLM A with LLM B. Earlier, in the "Setting" section of this paper, we justified why a rational actor would not attempt to impersonate LLM A with LLM B under the threat scenario (and imposed it as an explicit assumption in the threat scenario for good measure).

Therefore, we assert that it is justified to assume that the actor behind LLM B does not deliberately attempt to force $c \in S$.

A final potential objection to this axiom is the possibility of LLM B being a derived version of LLM A, whose implementation difference may be so negligible in principle that $c \in S$ becomes a real possibility with nonzero probability. However, our threat scenario explicitly excluded the possibility of LLM B being a fine-tune or some other derived version of LLM A when LLM A ≠ LLM B.

Thus, under the threat setting, when LLM B ≠ LLM A, we assert that LLM B should be viewed as being drawn from a non-adversarial distribution of independently-trained "potential" LLMs. This, combined with the measure 0 size of S, concludes our argument for Axiom 1 being a justified modeling assumption for the threat model.

A note about interpreting π:
The probability measure π in Axiom 1 may be interpreted either in a Bayesian sense (as a prior over potential configurations of LLM B) or in a frequentist sense (as a data-generating distribution over the configurations being created and encountered in practice). Our results are agnostic to which interpretation is used.

Having established Axiom 1, let us now continue with the proofs of the properties of the estimator algorithm and the hypothesis test.

Property 1 of the Estimator Algorithm: $\phi(g_1, ..., g_m)$ is a convergent statistic as $N^{(1)}, ..., N^{(w)} \to \infty$.
Proof: Lemma 2.   □

Property 2 of the Estimator Algorithm: If LLM A = LLM B, this $\phi(g_1, ..., g_m)$ converges to C. Else, it converges to a non-C value with probability 1.
Proof:
The estimator algorithm already assumes that $\phi(g_1, ..., g_m)$ converges to C when LLM A = LLM B.

Next, assume LLM A ≠ LLM B. We wish to show that the probability of $\phi(g_1, ..., g_m)$ converging to C is 0. By lemma 2, $\phi(g_1, ..., g_m)$ converges to $\phi(c_1, ..., c_m)$. Thus, we must show that the probability of $\phi(c_1, ..., c_m) = C$ is 0. $\phi(c_1, ..., c_m) = C$ is equivalent to $c \in S = \left\{ x \in \mathbb{R}^m : \phi(x) = C \right\}$. By axiom 1, the probability that $c \in S$ is 0.   □

We now prove the properties of the hypothesis test.

Property 1 of the hypothesis test: If H_0 if true, with probability 1, the test does not reject H_0 as $N^{(1)},..., N^{(w)} \rightarrow \infty$.

Proof:

Let us assume that H_0 is true (i.e. that LLM A = LLM B). Let us then establish some facts:

Fact 1: $\phi(g_1,..., g_m) \rightarrow C$ as $N^{(1)},..., N^{(w)} \rightarrow \infty$ (by assumption)

Fact 2: $\phi(g_1,..., g_m)$ approaches a normal distribution as $N^{(1)},..., N^{(w)} \rightarrow \infty$ (by Theorem 16, which is presented later in the paper)

Fact 3: $\sigma(\phi(g_1,.., g_m))$ converges to 0 in $\Theta\left(\sqrt{\sum_{r=1}^{w} \frac{1}{N^{(r)}}}\right)$ as $N^{(1)},..., N^{(w)} \rightarrow \infty$. (a direct consequence of Theorem 17, which is presented later in the paper)

Fact 4: Let $X \sim N\left(0, \sigma^2\right)$. Fix a percentile level $p \in (0, 1)$. Let $c_p(\sigma) = inf\{x : P(X \leqslant x) \geqslant p\}$ be the p-quantile of X. Then $c_p = z_p \sigma$ where $z_p = \Phi^{-1}(p)$ and Φ is the standard normal CDF.

Proof of Fact 4:

$P(X \leqslant x) = P\left(Z \leqslant \frac{x}{\sigma}\right)$. Thus, $P(X \leqslant c_p(\sigma)) = p$ (by definition) is equivalent to $P\left(Z \leqslant \frac{c_p(\sigma)}{\sigma}\right) = p$. By definition, we then have $\frac{c_p(\sigma)}{\sigma} = \Phi^{-1}(p) \implies c_p(\sigma) = z_p \sigma$. This completes the proof of fact 4.

Fact 5: Fix a percentile level $p \in (0, 1)$. Then $c_p(\sigma)$ converges to 0 in $\Theta(\sigma)$ (direct consequence of fact 4).

From theorem 17 and theorem 19, we have that as $N^{(1)},..., N^{(w)} \rightarrow \infty$, the standard deviation of $\phi(g_1,..., g_m)$ dominates its bias (see the comment following theorem 19). We may therefore consider bias to be negligible. As $N^{(1)},..., N^{(w)} \rightarrow \infty$, by facts 1-3, we may assume that $\phi(g_1,..., g_m) \sim N\left(C, \Theta\left(\sqrt{\sum_{r=1}^{w} \frac{1}{N^{(r)}}}\right)^2\right)$.

To prove property 1, it suffices to show that any p-quantile of $\phi(g_1,..., g_m)$ converges to C faster than $C \pm \epsilon\left(N^{(1)},..., N^{(w)}\right)$ does as $N^{(1)},..., N^{(w)} \rightarrow \infty$. Or, equivalently, it suffices to show that any p-quantile c_p of $\phi(g_1,..., g_m) - C \sim N\left(0, \Theta\left(\sqrt{\sum_{r=1}^{w} \frac{1}{N^{(r)}}}\right)^2\right)$ converges to 0

faster than $\pm\epsilon\left(N^{(1)},...,N^{(w)}\right)$ does as $N^{(1)},...,N^{(w)}\to\infty$.

Fact 5 tells us that any p-quantile c_p of $\phi(g_1,...,g_m)-C$ converges to 0 with

$$\Theta\left(\sqrt{\sum_{r=1}^{w}\frac{1}{N^{(r)}}}\right)$$ speed as $N^{(1)},...,N^{(w)}\to\infty$.

Since our hypothesis test enforces $\dfrac{\epsilon\left(N^{(1)},...,N^{(w)}\right)}{\sqrt{\sum_{r=1}^{w}1/N^{(r)}}}\to\infty$ as $N^{(1)},...,N^{(w)}\to\infty$, indeed, we

have that $\pm\epsilon\left(N^{(1)},...,N^{(w)}\right)$ converges to 0 slower than $\sqrt{\sum_{r=1}^{w}1/N^{(r)}}$ and therefore c_p too

as $N^{(1)},...,N^{(w)}\to\infty$. This completes the proof of property 1 of the hypothesis test. $\square$

Property 2 of the hypothesis test: If H_1 is true, with probability 1, the test rejects H_0 as $N^{(1)},...,N^{(w)}\to\infty$.

Proof:

Assume H_1 is true (i.e. LLM A $\neq$ LLM B). By property 2 of the estimator algorithm, we have

that $\phi(g_1,...,g_m)$ converges to a value $C'\neq C$ with probability 1. Let $m=\dfrac{|C-C'|}{2}$ be the

half distance between C' and C. Because $\phi(g_1,...,g_m)\to C'$ as $N^{(1)},...,N^{(w)}\to\infty$,

eventually, $\phi(g_1,...,g_m)$ will remain in $\mathbb{R}\setminus[C-m,C+m]$. Note that the hypothesis test

imposes $\epsilon\left(N^{(1)},...,N^{(w)}\right)\downarrow 0$ as $N^{(1)},...,N^{(w)}\to\infty$. Thus, eventually we will have

$\left[C-\epsilon\left(N^{(1)},...,N^{(w)}\right),C+\epsilon\left(N^{(1)},...,N^{(w)}\right)\right]\subsetneq[C-m,c+m]$ as $N^{(1)},...,N^{(w)}\to\infty$. Since

the decision rule is to reject H_0 if $\phi(g_1,...,g_m)\in\mathbb{R}\setminus[C-m,c+m]$, the hypothesis test will

eventually reject H_0 as $N^{(1)},...,N^{(w)}\to\infty$. $\square$

Final remarks for this section

We've produced *consistent* tests for LLM ownership identity under our setting and assumptions. In this sense, these tests are theoretically "perfect" deciders of whether LLM A = LLM B under our threat scenario. Our general test uses g_j's as their fundamental building blocks, and we've explicitly demonstrated that one can create concrete tests from *est* and *est2*, which themselves are built from these g_j's.

Robustness

There is another point that we wish to bring attention to. When we built the $\phi(g_1,...,g_m)$'s in the "Specific Examples" subsection, the contexts C_j and strings $a_{1,j},...,a_{k_j,j},b_{1,j},...,b_{k_j,j}$ we used to construct them were constrained but otherwise arbitrary choices. This points to the

fact that there are a massive class of possible $\phi(g_1,...,g_m)$ estimators that each converge to their corresponding C if LLM A = LLM B, but converge to a non-C value otherwise. The key takeaway is this - there is a vast set of "theoretically perfect" tests for deciding whether or not LLM A = LLM B that can be constructed. The quantity of this redundancy makes testing whether LLM A = LLM B exceptionally robust.

Generalization to other threat scenarios: Impersonation Attacks
How well can this test hold up under other threat scenarios? Consider the scenario where LLM A $\neq$ LLM B, but the actor behind LLM B is attempting to cause our tests to erroneously detect LLM B as LLM A. In other words, consider the scenario where an impersonation attack were to happen.

For our tests to become asymptotically invalid, it is necessary for Axiom 1 to fail. In other words, such an adversary would need to force c to land on the measure-0 set S. Furthermore, if multiple tests are created like in our above robustness discussion, such an adversary would need to force c to collapse onto a measure-0 set S for every single one of these tests. It is worth asking if it is even practically feasible for such an adversary to accomplish this. This motivates our conjecture that it is "hard" for an adversary with a different LLM to impersonate LLM A under our testing scheme, and we encourage further research to explore this direction.

Asymptotic Analysis: Independent g_j's

We begin our analysis of the asymptotic properties of $\phi(g_1,...,g_m)$ by considering a well-behaved but useful case: when g_j's are independent. In this section, each g_j is assumed to be constructed from an independent, self-contained experiment involving running LLM B many times on context C_j and recording number of occurrences of $a_{1,j},..., a_{k_j,j}, b_{1,j},..., b_{k_j,j}$.

We will be able to prove precise formulas instead of results written in the more abstract big-O or big-Θ notation found in the general case (See the next section: "Asymptotic Analysis: Non-independent g_j's (The General Case)"). This will allow us to analyze the actual coefficients of the terms, which then allows us to extract higher-resolution insights that are amenable to practical-case analysis, where coefficients often matter. Much of the ideas, tools, techniques, and intuition developed in this section carry forward and help guide the analysis of the generic non-independent case in the final section of this paper.

In this section, we will prove several core results for the independent case, then discuss how these results can be used to design $\phi(g_1,...,g_m)$ estimators that can converge within a 'reasonable' or 'practical' number of calls to LLM B. We will also provide a few examples of variance computations for a couple of concrete estimators of interest.

We will adopt the language and notation of the previous section in this section. When we consider each g_j to be constructed from an independent, self-contained experiment, we are saying that the j's and the experiments ($\{E_r : r = 1 : w\}$) have a 1-1 correspondence. Each $j = 1 : m$ corresponds to a unique $r = 1 : w$ and vice versa. This implies $w = m$. This also implies that the N_j's and $N^{(r)}$'s are identical up to reordering. It is therefore redundant to speak in the language of experiments in this independent g_j setting. For bookkeeping simplicity, we will speak in the language of j-indexed concepts rather than in the language of experiments (r-indexed concepts) for the rest of this section. Just keep in mind that they are interchangeable concepts in this setting.

Let us now state our assumptions for this section clearly:

- Throughout this section, $\phi(x_1,..., x_m)$ will be an arbitrary smooth function unless otherwise stated.
- Throughout this section, we shall restrict our attention to the case where the g_j's are decoupled and independent. In other words, each g_j is constructed from an independent, self-contained experiment involving running LLM B many times on context C_j and recording number of occurrences of a_{1,j},...,a_{k_j,j},b_{1,j},...,b_{k_j,j}.
- Throughout this section, we will be assuming †† instead of † (which is a special case of ††; †† is a relaxation of †).
- Additional Regularity Condition (*): For each C_j, it is impossible for a_{1,j},...,a_{k_j,j},b_{1,j},...,b_{k_j,j} to be string completions of each other with respect to LLM B (practical case: a_{1,j},...,a_{k_j,j},b_{1,j},...,b_{k_j,j} are not substrings of each other).

Note: (*) imposes a nice regularity structure that will allow us to derive clean, precise variance formulas. In particular, due to (*) preventing overlapping generations among $a_{1,j},..., a_{k_j,j}, b_{1,j},..., b_{k_j,j}$ for each j, the $n_j(a, i)$'s and $n_j(b, i)$'s are components of a multinomial distribution for each j.

The Core Theorems
As it turns out, $\phi(g_1,..., g_m)$ and the g_j's have multiple desirable estimator properties:

Theorem 7: $g_1,..., g_m$ are independent.

Proof: By construction.   □

Theorem 8: g_j is a consistent estimator of c_j as $N_j \to \infty$.
Proof: This is a restatement of lemma 1. We've inserted it here for presentation.   □

Theorem 9: $var(g_j) \to \dfrac{1}{N_j}\left(\displaystyle\sum_{i=1}^{k_j}\dfrac{1}{p_j(a,i)} + \sum_{i=1}^{k_j}\dfrac{1}{p_j(b,i)}\right)$ as $N_j \to \infty$

Proof: Deferred for later (★★). □

Comment: g_j's variance converges in $O(1/N_j)$ speed, which matches the best possible parametric decay rate $O(1/N_j)$ for the variance of an estimator satisfying standard regularity conditions.

Theorem 10: Each g_j is asymptotically a normal random variable as $N_j \to \infty$.

Proof: Deferred for later (★★). □

Theorem 11: $\phi(g_1,...,g_m)$ is a consistent estimator of $\phi(c_1,...,c_m)$ as $N_1,...,N_m \to \infty$

Proof: Theorem 8. □

Theorem 12: $var(\phi(g_1,...,g_m)) \to \displaystyle\sum_{j=1}^{m}\dfrac{1}{N_j}\left(\dfrac{\partial\phi}{\partial g_j}(c_1,...,c_m)\right)^2\left(\sum_{i=1}^{k_j}\dfrac{1}{p_j(a,i)} + \sum_{i=1}^{k_j}\dfrac{1}{p_j(b,i)}\right)$ as

$N_1,...,N_m \to \infty$.

Proof:

By Theorem 8, g_j is a consistent estimator of c_j, so as $N_1,...,N_m \to \infty$, we have by the delta method:

$$\phi(g_1,...,g_m) \approx \phi(c_1,...,c_m) + \sum_{j=1}^{m}\dfrac{\partial\phi}{\partial g_j}(c_1,...,c_m)(g_j - c_j)$$

$$\implies var(\phi(g_1,...,g_m)) \approx \sum_{j=1}^{m}\left(\dfrac{\partial\phi}{\partial g_j}(c_1,...,c_m)\right)^2 var(g_j)$$

$$= \sum_{j=1}^{m}\dfrac{1}{N_j}\left(\dfrac{\partial\phi}{\partial g_j}(c_1,...,c_m)\right)^2\left(\sum_{i=1}^{k_j}\dfrac{1}{p_j(a,i)} + \sum_{i=1}^{k_j}\dfrac{1}{p_j(b,i)}\right) \text{ (Theorem 9)} \qquad □$$

Comment: Since we know $var(\phi(g_1,...,g_m))$ is $O\left(\displaystyle\sum_{j=1}^{m}\dfrac{1}{N_j}\right)$ as $N_1,...,N_m \to \infty$, then taking

the N_j's to be fixed proportions of $N = \sum N_j$ (the total number of samples/calls/runs across all experiments) results in $O(1/N)$ variance decay for $\phi(g_1,...,g_m)$. This matches the best possible parametric decay rate $O(1/N)$ for the variance of an estimator satisfying standard

regularity conditions.

Theorem 13: $\phi(g_1,...,g_m)$ is asymptotically a normal random variable as $N_1,...,N_m \to \infty$.

Proof:
By Theorem 8, g_j is a consistent estimator of c_j, so as $N_1,...,N_m \to \infty$, we have by the delta method:

$$\phi(g_1,...,g_m) \approx \phi(c_1,...,c_m) + \sum_{j=1}^{m} \frac{\partial \phi}{\partial g_j}(c_1,...,c_m)(g_j - c_j)$$

Theorem 10 tells us that the g_j's are asymptotically normal and Theorem 7 tells us that they are independent. Then, the right hand side of the equation is an affine transformation of independent, asymptotically normal random variables. It is a well known fact that an affine transformation of independent normal random variables is normal. This completes the proof.

$\square$

Theorem 14: $Bias(\phi(g_1,...,g_m)) = E[\phi(g_1,...,g_m)] - \phi(c_1,...,c_m) = O\left(\sum_{j=1}^{m} 1/N_j\right)$

Proof: Deferred for later (★★).

$\square$

Comment: From theorem 12, we have that $\sigma(\phi(g_1,...,g_m))$ decays at rate $\Theta\left(\sqrt{\sum_{j=1}^{m} 1/N_j}\right)$,

which asymptotically dominates $Bias(\phi(g_1,...,g_m)) = O\left(\sum_{j=1}^{m} 1/N_j\right)$. In other words,

$Bias(\phi(g_1,...,g_m)) = o(\sigma(\phi(g_1,...,g_m))$. Combined with the fact that $\phi(g_1,...,g_m)$ is asymptotically normal, this allows us to treat bias as negligible ($= 0$) as $N_1,...,N_m \to \infty$ for the rest of this section.

Heuristic Analysis
As we've discussed before, $\phi(g_1,...,g_m)$ is an estimator that serves as the central tool in our tests for LLM ownership identity. Naturally, we desire to optimize its convergence speed. Since $\phi(g_1,...,g_m)$ is consistent and asymptotically normal, this problem reduces to minimizing its asymptotic variance.

Let us inspect the equation

$$var(\phi(g_1,...,g_m)) \to \sum_{j=1}^{m} \frac{1}{N_j}\left(\frac{\partial \phi}{\partial g_j}(c_1,...,c_m)\right)^2 \left(\sum_{i=1}^{k_j} \frac{1}{p_j(a,i)} + \sum_{i=1}^{k_j} \frac{1}{p_j(b,i)}\right)$$. It is crucial to note

that we do not, in practice, know the values of the $p_j(a, i)$'s and $p_j(b, i)$'s. After all, they are based on black-box LLM B's internal configuration, which we do not assume access to. However, this does not mean we cannot extract heuristic insights from the structure of this equation.

First, note that heuristically speaking, $var(\phi(g_1,...,g_m))$ seems to increase as m and the k_j's increase (assuming all else being held equal, particularly the values of $\frac{\partial \phi}{\partial g_j}(c_1,..., c_m)$). It therefore seems reasonable to try to minimize m and the k_j's when building the estimator $\phi(g_1,..., g_m)$, as well as maximize the $p_j(a, i)$ and $p_j(b, i)$'s.

Note that est and $est2$ themselves are each built from 2 instances of g_j. Therefore, any estimator $\phi(g_1,..., g_m)$ built using est or $est2$ as one of its components has $m \geqslant 2$. Naive Method I and Naive Method II each has $m = 4$.

For maximizing the $p_j(a, i)$ and $p_j(b, i)$'s, again, we do not have access to LLM B's internals, so we cannot do this for certain. However, we can use our human semantic intuition about LLM outputs to intelligently guess high-probability strings $a_{i,j}, b_{i,j}$ for LLM B on context C_j. For example, if C_j were:

```
<|begin_of_text|>
<|user|>
Output a random binary digit. Do nothing else.
<|assistant|>
```

Then we might choose $a_{i,j}$ to be "0" and $b_{i,j}$ to be "1".

We summarize our heuristic takeaways with 3 rules of thumb for optimizing the convergence speed of $\phi(g_1,..., g_m)$:
1. Try to make m small.
2. Try to make the k_j's small (aim for $k_j = 1$ for each $j = 1 : m$).
3. Try to make each $p_j(a, i)$ and $p_j(b, i)$ large by intelligently choosing strings $a_{i,j}$ and $b_{i,j}$ for context C_j.

Optimizing convergence speed under a fixed number-of-runs budget
Consider again the equation

$$var(\phi(g_1,...,g_m)) \rightarrow \sum_{j=1}^{m} \frac{1}{N_j} \left(\frac{\partial \phi}{\partial g_j}(c_1,..., c_m) \right)^2 \left(\sum_{i=1}^{k_j} \frac{1}{p_j(a, i)} + \sum_{i=1}^{k_j} \frac{1}{p_j(b, i)} \right). \text{ One thing we}$$

haven't addressed yet is how to choose the N_j's to maximize convergence speed. In

practice, our number-of-runs budget is constrained as $N_1 + ... + N_m = N$ for some fixed N. A method for computing optimal values of $N_1,..., N_m$ given a compute budget of N runs is given below, which assumes that we have the values of the $p_j(a, i)$'s and $p_j(b, i)$'s (Again, in practice, this is a nontrivial assumption. We will discuss a heuristic method for computing optimal values of N_j in the "Heuristic Proportion Selection" subsection).

Let us introduce a new shorthand:

- Let $q_j = \left(\dfrac{\partial \phi}{\partial g_j}(c_1,..., c_m) \right)^2 \left(\displaystyle\sum_{i=1}^{k_j} \dfrac{1}{p_j(a, i)} + \sum_{i=1}^{k_j} \dfrac{1}{p_j(b, i)} \right)$. Note that q_j is a nonnegative constant.

Theorem 15: Assume $N_1,..., N_m \to \infty$. Given a run budget of $N_1 + ... + N_m = N$,

$N_j^* \approx \dfrac{N\sqrt{q_j}}{\sum_{l=1}^{m} \sqrt{q_l}}$ becomes the optimal choice of N_j to maximize convergence speed, with

$var(\phi(g_1,..., g_m)) \to \dfrac{\left(\sum_{j=1}^{m} \sqrt{q_j} \right)^2}{N}$ in this case.

Proof:

By theorem 12, we have that $var(\phi(g_1,..., g_m)) \to \displaystyle\sum_{j=1}^{m} \dfrac{q_j}{N_j}$. We apply Lagrange multipliers to the problem of optimizing $\displaystyle\sum_{j=1}^{m} \dfrac{q_j}{N_j}$ under the constraint $N_1 + ... + N_m = N$. Note that we intentionally did not include the $N_1,..., N_m \geqslant 0$ constraints. We will later find that the optimizer actually satisfies those constraints anyway.

Lagrangian: $\mathcal{L}(N_1,..., N_m; \lambda) = \displaystyle\sum_{j=1}^{m} \dfrac{q_j}{N_j} + \lambda \left(\sum_{j=1}^{m} N_j - N \right)$.

$\dfrac{\partial \mathcal{L}}{\partial N_j} = 0 \implies \lambda = \dfrac{q_j}{N_j^2} \implies N_j = \sqrt{\dfrac{q_j}{\lambda}}$ (2)

By the constraint $\sum N_j = N$, we have:

$\displaystyle\sum_{j=1}^{m} \sqrt{\dfrac{q_j}{\lambda}} = N$

$$\implies \lambda = \frac{\left(\Sigma_{j=1}^{m}\sqrt{q_j}\right)^2}{N^2} \quad (3)$$

Plugging (3) into (2) to get $N_j^* = \dfrac{N\sqrt{q_j}}{\Sigma_{l=1}^{m}\sqrt{q_l}}.$

Observe that $N_j^* > 0$, which satisfies the implicit $N_1,..., N_m \geqslant 0$ constraints. Furthermore, $N_j^* \to \infty$ when $N_1 + ... + N_m = N \to \infty$. Therefore, this optimizer is consistent with our

$N_1,..., N_m \to \infty$ assumption. We conclude that $N_j^* \approx \dfrac{N\sqrt{q_j}}{\Sigma_{l=1}^{m}\sqrt{q_l}}$ is our desired optimizer.

Plugging the N_j^*'s back in, we get:

$$var(\phi(g_1,..., g_m)) \to \sum_{j=1}^{m}\frac{q_j}{N_j^*}$$

$$= \sum_{j=1}^{m}\frac{q_j}{\dfrac{N\sqrt{q_j}}{\Sigma_{l=1}^{m}\sqrt{q_l}}}$$

$$= \frac{\Sigma_{l=1}^{m}\sqrt{q_l}}{N}\sum_{j=1}^{m}q_j$$

$$= \frac{\left(\Sigma_{j=1}^{m}\sqrt{q_j}\right)^2}{N}$$

$\square$

Comment: A key structural insight from this theorem is that the optimal values $N_j^*, j = 1 : m$ are fixed proportions of N which don't depend on N. Thus, it often makes more practical sense to solve for the proportions rather than solve for N_j^*'s themselves; solving for proportions will allow us to reuse them across different N's.

Corollary 5: Assume $N_1,..., N_m \to \infty$. Given a run budget of $N_1 + ... + N_m = N$, we have:

1. $\dfrac{\partial}{\partial N_j}var(\phi(g_1,..., g_m)) \to -\dfrac{\left(\Sigma_{j=1}^{m}\sqrt{q_j}\right)^2}{N^2}$ (where we assume $N_j = \dfrac{N\sqrt{q_j}}{\Sigma_{l=1}^{m}\sqrt{q_l}}$ for $j = 1 : m$)

2. Let $F(N) = min_{\sum N_j = N} var(\phi(g_1,..., g_m))$. Then $\dfrac{\partial}{\partial N}F(N) \to -\dfrac{\left(\sum_{j=1}^{m} \sqrt{q_j}\right)^2}{N^2}$

Proof:

Revisiting the Lagrangian of the problem of optimizing $\displaystyle\sum_{j=1}^{m} \dfrac{q_j}{N_j}$ under the constraint

$N_1 + ... + N_m = N$, we have: $\mathcal{L}(N_1,..., N_m; \lambda) = \displaystyle\sum_{j=1}^{m} \dfrac{q_j}{N_j} + \lambda\left(\sum_{j=1}^{m} N_j - N\right)$.

$\dfrac{\partial \mathcal{L}}{\partial N_j} = 0 \implies \dfrac{\partial}{\partial N_j} var(\phi(g_1,..., g_m)) \approx \dfrac{\partial}{\partial N_j}\left(\displaystyle\sum_{j=1}^{m} \dfrac{q_j}{N_j}\right) = -\lambda$

Plugging in $\lambda = \dfrac{\left(\sum_{j=1}^{m} \sqrt{q_j}\right)^2}{N^2}$ from equation (3) proves 1.

2. is a direct consequence of the fact that $var(\phi(g_1,..., g_m))$ asymptotically converges to

$\displaystyle\sum_{j=1}^{m} \dfrac{q_j}{N_j}$ and the Envelope Theorem, a well-known result in constrained optimization   □

<u>Heuristic Proportion Selection</u>

First, define $s_j := \dfrac{\sqrt{q_j}}{\sum_{l=1}^{m} \sqrt{q_l}}$ for $j = 1 : m$. The core practical issue we continue to face is that
the value of s_j depends on our knowledge of the $p_j(a, i)$'s and $p_j(b, i)$'s, which requires knowledge of black-box LLM B's internal configuration (not assumed). Below, we outline a heuristic approach for estimating good values of s_j for optimizing the convergence speed of $\phi(g_1,..., g_m)$.

Definition: Let p be a vector whose entries are all of the $p_j(a, i)$'s and $p_j(b, i)$'s. In particular, p is a vector in $\mathbb{R}^{\sum 2k_j}$.

A Heuristic Monte Carlo method:
Goal: Find a convergence-speed-optimal proportion vector $s = \langle s_1,..., s_m\rangle$ representing the proportions of N used to compute the N_j^*'s.
Steps:

1. Viewing the components of p as random variables, guess a prior distribution π for p (e.g. guess a plausible range for each $p_j(a, i)$ and each $p_j(b, i)$ to independently uniformly randomly sample from)
2. Repeatedly sample p from π and compute $s = \langle s_1, .., s_m \rangle$ for each sample. Note that the computations of the s's are independent of N.
3. Take the mean of the s's to get an approximation for $E_{p \sim \pi}[s]$. The final vector will be the proportions of N that we can use to compute N_j^*'s.

A more sophisticated method for creating a generative π distribution:
1. Procure a list of white-box LLMs (which may include LLM A and open-source/open-weight models). Call these LLMs $M_1, ..., M_l$.
2. For each standard token-distribution-modifying decoding parameter, guess a pragmatic range that it lives in.
3. A LLM configuration Θ is defined as a choice of LLM and exact values for its standard token-distribution-modifying decoding parameters. Sample Θ by uniformly randomly selecting an LLM from $\{M_1, ..., M_l\}$ and standard token-distribution-modifying decoding parameters from their corresponding ranges.
4. For each Θ sampled, compute the $p_j(a, i)$'s and $p_j(b, i)$'s associated with it.

The overall idea is that although we don't know LLM B's exact $p_j(a, i)$'s and $p_j(b, i)$'s, we can use *other* LLM configurations to compute noisy approximations for them.

<u>**Final Estimator Pipeline**</u>

Let us aggregate our previous ideas into a full pipeline for $\phi(g_1, ..., g_m)$ estimator design and estimation.

Note: There is a classic optimization/pedagogical clarity tradeoff when it comes to presenting algorithms. We have chosen to present this pipeline in an illustrative, conceptually clear way. However, note that for a practical deployment, there will be redundant computations in the pseudocode below if it is implemented exactly as-is.

Goal: Produce an estimator $\phi(g_1, ..., g_m)$ that converges quickly.

Steps:
1. Using our 3 heuristic rules of thumb, construct a list of candidate estimators $\phi(g_1, ..., g_m)$. Do not yet instantiate the values of the N_j's.
2. For each candidate estimator $\phi(g_1, ..., g_m)$, compute a proportion vector $s_{\phi(g_1, ..., g_m)} = \langle s_1, .., s_m \rangle$ using heuristic proportion selection.
3. For each candidate estimator $\phi(g_1, ..., g_m)$ and its corresponding proportion vector proportion vector $s_{\phi(g_1, ..., g_m)} = \langle s_1, .., s_m \rangle$, compute

$$v_{\phi(g_1,...,g_m)} = \sum_{j=1}^{m} \frac{1}{s_j} \left(\frac{\partial \phi}{\partial g_j}(c_1,...,c_m) \right)^2 \left(\sum_{i=1}^{k_j} \frac{1}{p_j(a,i)} + \sum_{i=1}^{k_j} \frac{1}{p_j(b,i)} \right).$$

4. Choose the candidate estimator $\phi(g_1,...,g_m)$ and corresponding proportion vector $s_{\phi(g_1,...,g_m)} = \langle s_1,..,s_m \rangle$ with the lowest $v_{\phi(g_1,...,g_m)}$.

Note: $s_{\phi(g_1,...,g_m)}$ is a heuristically chosen proportion vector, representing our best guess of the proportions of N that each N_j^* must take to minimize variance/convergence speed of $\phi(g_1,...,g_m)$. Based on theorem 12 and theorem 15, $v_{\phi(g_1,...,g_m)}$ is the variance approximation of $\phi(g_1,...,g_m)$ using $s_{\phi(g_1,...,g_m)}$, up to the factor $1/N$.

Having chosen $\phi(g_1,...,g_m)$ and $s_{\phi(g_1,...,g_m)} = \langle s_1,..,s_m \rangle$, we now have a natural estimation algorithm:

Goal: Estimate the limit of $\phi(g_1,...,g_m)$.
Steps:
1. Iteratively increase the number of total observations N while maintaining $N_1,...,N_m$ to have the fixed relative proportions $s_{\phi(g_1,...,g_m)} = \langle s_1,..,s_m \rangle$.
2. Apply an early stopping rule.

<u>Variance Calculation Examples</u>
We'll now explore specific variance examples that may of be interest to the reader and which will serve to elucidate some of the previous results.

<u>The estimator est</u>

For the rest of this "The estimator est" subsection, we are assuming that contexts C_1, C_2 and strings $a_1,...,a_k$ and $b_1,...,b_k$ satisfy the (†) regularity conditions, in addition to the prevailing assumptions being made throughout the entire "Asymptotic Analysis: Independent g_j's" section.

Recall est: $est(C_1; C_2; a_1,..., a_k; b_1,..., b_k) = \dfrac{\Delta z_1 - \Delta z_2}{ln\left(\prod_{i=1}^{k} \frac{n_1(a,i)}{n_1(b,i)} \right) - ln\left(\prod_{i=1}^{k} \frac{n_2(a,i)}{n_2(b,i)} \right)}$

We can rewrite this as $est(C_1; C_2; a_1,..., a_k; b_1,..., b_k) = \phi(g_1, g_2)$ where

$$g_1 = ln\left(\prod_{i=1}^{k} \frac{n_1(a,i)}{n_1(b,i)} \right), g_2 = ln\left(\prod_{i=1}^{k} \frac{n_2(a,i)}{n_2(b,i)} \right), \phi(g_1, g_2) = \frac{\Delta z_1 - \Delta z_2}{g_1 - g_2}.$$

Note: $est(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ converges to T when LLM A = LLM B, which is an unknown constant. Therefore, $est(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ cannot be used as the estimator in the estimator algorithm. Nevertheless, it can be expressed in the form of $\phi(g_1, g_2)$ and the theorems in the previous section can still be used to analyze it.

We then have that theorem 11, theorem 12, and theorem 13 hold, so we know that $est(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ has desirable properties like consistency, asymptotic normality, and an asymptotic approximation for its variance. Let's compute its asymptotic variance.

By Theorem 12, we have that

$$var(est(C_1; C_2; a_1,..., a_k; b_1,..., b_k)) \to \sum_{j=1}^{2} \frac{1}{N_j} \left(\frac{\partial \phi}{\partial g_j}(c_1, c_2) \right)^2 \left(\sum_{i=1}^{k} \frac{1}{p_j(a, i)} + \sum_{i=1}^{k} \frac{1}{p_j(b, i)} \right) \text{ as}$$

$N_1, N_2 \to \infty$.
We have:

$$\frac{\partial \phi}{\partial g_1}(c_1, c_2) = -\frac{\Delta z_1 - \Delta z_2}{(c_1 - c_2)^2} \text{ and } \frac{\partial \phi}{\partial g_2}(c_1, c_2) = \frac{\Delta z_1 - \Delta z_2}{(c_1 - c_2)^2}.$$

Thus, we have:

$$\sum_{j=1}^{2} \frac{1}{N_j} \left(\frac{\partial \phi}{\partial g_j}(c_1, c_2) \right)^2 \left(\sum_{i=1}^{k} \frac{1}{p_j(a, i)} + \sum_{i=1}^{k} \frac{1}{p_j(b, i)} \right)$$

$$= \frac{(\Delta z_1 - \Delta z_2)^2}{(c_1 - c_2)^4} \left(\frac{1}{N_1} \left(\sum_{i=1}^{k} \frac{1}{p_1(a, i)} + \sum_{i=1}^{k} \frac{1}{p_1(b, i)} \right) + \frac{1}{N_2} \left(\sum_{i=1}^{k} \frac{1}{p_2(a, i)} + \sum_{i=1}^{k} \frac{1}{p_2(b, i)} \right) \right)$$

This gives us:

Corollary 6:
$$var(est(C_1; C_2; a_1,..., a_k; b_1,..., b_k))$$

$$\to \frac{(\Delta z_1 - \Delta z_2)^2}{(c_1 - c_2)^4} \left(\frac{1}{N_1} \left(\sum_{i=1}^{k} \frac{1}{p_1(a, i)} + \sum_{i=1}^{k} \frac{1}{p_1(b, i)} \right) + \frac{1}{N_2} \left(\sum_{i=1}^{k} \frac{1}{p_2(a, i)} + \sum_{i=1}^{k} \frac{1}{p_2(b, i)} \right) \right) \text{ as}$$

$N_1, N_2 \to \infty$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

Recall that when LLM A = LLM B, we have that $c_j = \dfrac{\Delta z_j + \Delta b}{T}$. This gives us:

Corollary 7: If LLM A = LLM B,

$var(est(C_1; C_2; a_1,..., a_k; b_1,..., b_k))$

$$\rightarrow \frac{T^4}{(\Delta z_1 - \Delta z_2)^2}\left[\frac{1}{N_1}\left(\sum_{i=1}^{k}\frac{1}{p_1(a,i)} + \sum_{i=1}^{k}\frac{1}{p_1(b,i)}\right) + \frac{1}{N_2}\left(\sum_{i=1}^{k}\frac{1}{p_2(a,i)} + \sum_{i=1}^{k}\frac{1}{p_2(b,i)}\right)\right]$$

Proof:

$$\frac{(\Delta z_1 - \Delta z_2)^2}{(c_1 - c_2)^4} = \frac{(\Delta z_1 - \Delta z_2)^2}{\left(\frac{\Delta z_1 + \Delta b}{T} - \frac{\Delta z_2 + \Delta b}{T}\right)^4} = \frac{T^4}{(\Delta z_1 - \Delta z_2)^2}$$ □

Heuristic Analysis:

Looking at the case of LLM A = LLM B gives us some further heuristic intuition for boosting the convergence speed of est. In particular, focus on the $(\Delta z_1 - \Delta z_2)^2$ term. Variance is inversely proportional to it. Recalling our definitions of $\Delta z_1, \Delta z_2$, we see that

$$|\Delta z_1 - \Delta z_2| = \left|\left(\sum_{i=1}^{k} z_{a_i,1} - \sum_{i=1}^{k} z_{b_i,1}\right) - \left(\sum_{i=1}^{k} z_{a_i,2} - \sum_{i=1}^{k} z_{b_i,2}\right)\right|.$$

Fortunately, because we have white-box access to LLM A and therefore all $z_{a_i,j}$'s and $z_{b_i,j}$'s are knowable to us, we can cycle through various candidates for C_1, C_2 and $a_1,..., a_k, b_1,..., b_k$ until our computed value $|\Delta z_1 - \Delta z_2|$ is reasonably large.

To reduce time in producing candidates for C_1, C_2, and $a_1,..., a_k, b_1,..., b_k$, we may adopt the following heuristic: "Pick C_1, C_2 and $a_1,..., a_k, b_1,..., b_k$ such that $a_1,..., a_k$ are more likely to be generated than $b_1,..., b_k$ on context C_1 and $b_1,..., b_k$ are more likely to be generated than $a_1,..., a_k$ on context C_2 (or vice versa)."

The estimator est2

For the rest of this "The estimator est" subsection, we are assuming that contexts C_1, C_2 and strings $a_1,...,a_k$ and $b_1,...,b_k$ satisfy the (†) regularity conditions, in addition to the prevailing assumptions being made throughout the entire "Asymptotic Analysis: Independent g_j's" section.

Recall $est2$: $est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k) = \dfrac{ln\left(\prod_{i=1}^{k}\frac{n_2(a,i)}{n_2(b,i)}\right)\Delta z_1 - ln\left(\prod_{i=1}^{k}\frac{n_1(a,i)}{n_1(b,i)}\right)\Delta z_2}{ln\left(\prod_{i=1}^{k}\frac{n_1(a,i)}{n_1(b,i)}\right) - ln\left(\prod_{i=1}^{k}\frac{n_2(a,i)}{n_2(b,i)}\right)}$

We can rewrite this as $est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k) = \phi(g_1, g_2)$ where

$$g_1 = ln\left(\prod_{i=1}^{k} \frac{n_1(a, i)}{n_1(b, i)}\right), g_2 = ln\left(\prod_{i=1}^{k} \frac{n_2(a, i)}{n_2(b, i)}\right), \phi(g_1, g_2) = \frac{g_2 \Delta z_1 - g_1 \Delta z_2}{g_1 - g_2}.$$

Note: $est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ converges to Δb when LLM A = LLM B, which is an unknown constant. Therefore, $est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ cannot be used as the estimator in the estimator algorithm. Nevertheless, it can be expressed in the form of $\phi(g_1, g_2)$ and the theorems in the previous section can still be used to analyze it.

As usual, we then have that theorem 11, theorem 12, and theorem 13 hold, so we know that $est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k)$ has desirable properties like consistency, asymptotic normality, and an asymptotic approximation for its variance. Let's compute its asymptotic variance.

By Theorem 12, we have that

$$var(est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k)) \rightarrow \sum_{j=1}^{2} \frac{1}{N_j}\left(\frac{\partial \phi}{\partial g_j}(c_1, c_2)\right)^2 \left(\sum_{i=1}^{k} \frac{1}{p_j(a, i)} + \sum_{i=1}^{k} \frac{1}{p_j(b, i)}\right)$$ as

$N_1, N_2 \rightarrow \infty$.
We have:

$$\frac{\partial \phi}{\partial g_1}(c_1, c_2) = -\frac{c_2(\Delta z_1 - \Delta z_2)}{(c_1 - c_2)^2} \text{ and } \frac{\partial \phi}{\partial g_2}(c_1, c_2) = \frac{c_1(\Delta z_1 - \Delta z_2)}{(c_1 - c_2)^2}.$$

Thus, we have:

$$\sum_{j=1}^{2} \frac{1}{N_j}\left(\frac{\partial \phi}{\partial g_j}(c_1, c_2)\right)^2 \left(\sum_{i=1}^{k} \frac{1}{p_j(a, i)} + \sum_{i=1}^{k} \frac{1}{p_j(b, i)}\right)$$

$$= \frac{(\Delta z_1 - \Delta z_2)^2}{(c_1 - c_2)^4}\left[\frac{c_2^2}{N_1}\left(\sum_{i=1}^{k} \frac{1}{p_1(a, i)} + \sum_{i=1}^{k} \frac{1}{p_1(b, i)}\right) + \frac{c_1^2}{N_2}\left(\sum_{i=1}^{k} \frac{1}{p_2(a, i)} + \sum_{i=1}^{k} \frac{1}{p_2(b, i)}\right)\right]$$

This gives us:

Corollary 8:
$$var(est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k))$$
$$\rightarrow \frac{(\Delta z_1 - \Delta z_2)^2}{(c_1 - c_2)^4}\left[\frac{c_2^2}{N_1}\left(\sum_{i=1}^{k} \frac{1}{p_1(a, i)} + \sum_{i=1}^{k} \frac{1}{p_1(b, i)}\right) + \frac{c_1^2}{N_2}\left(\sum_{i=1}^{k} \frac{1}{p_2(a, i)} + \sum_{i=1}^{k} \frac{1}{p_2(b, i)}\right)\right]$$ as

$$N_1, N_2 \to \infty$$

□

Recall that when LLM A = LLM B, we have that $c_j = \dfrac{\Delta z_j + \Delta b}{T}$. This gives us:

Corollary 9: If LLM A = LLM B,

$var(est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k))$

$$\to \frac{T^2}{(\Delta z_1 - \Delta z_2)^2}\left[\frac{(\Delta z_2 + \Delta b)^2}{N_1}\left(\sum_{i=1}^{k}\frac{1}{p_1(a, i)} + \sum_{i=1}^{k}\frac{1}{p_1(b, i)}\right) + \frac{(\Delta z_1 + \Delta b)^2}{N_2}\left(\sum_{i=1}^{k}\frac{1}{p_2(a, i)} + \sum_{i=1}^{k}\frac{1}{p_2(b, i)}\right)\right]$$

Proof:

$$\frac{(\Delta z_1 - \Delta z_2)^2}{(c_1 - c_2)^4}\left[\frac{c_2^2}{N_1}\left(\sum_{i=1}^{k}\frac{1}{p_1(a, i)} + \sum_{i=1}^{k}\frac{1}{p_1(b, i)}\right) + \frac{c_1^2}{N_2}\left(\sum_{i=1}^{k}\frac{1}{p_2(a, i)} + \sum_{i=1}^{k}\frac{1}{p_2(b, i)}\right)\right]$$

$$= \frac{(\Delta z_1 - \Delta z_2)^2}{\left(\frac{\Delta z_1 + \Delta b}{T} - \frac{\Delta z_2 + \Delta b}{T}\right)^4}\left[\frac{\left(\frac{\Delta z_2 + \Delta b}{T}\right)^2}{N_1}\left(\sum_{i=1}^{k}\frac{1}{p_1(a, i)} + \sum_{i=1}^{k}\frac{1}{p_1(b, i)}\right) + \frac{\left(\frac{\Delta z_1 + \Delta b}{T}\right)^2}{N_2}\left(\sum_{i=1}^{k}\frac{1}{p_2(a, i)} + \sum_{i=1}^{k}\frac{1}{p_2(b, i)}\right)\right]$$

$$= \frac{T^2}{(\Delta z_1 - \Delta z_2)^2}\left[\frac{(\Delta z_2 + \Delta b)^2}{N_1}\left(\sum_{i=1}^{k}\frac{1}{p_1(a, i)} + \sum_{i=1}^{k}\frac{1}{p_1(b, i)}\right) + \frac{(\Delta z_1 + \Delta b)^2}{N_2}\left(\sum_{i=1}^{k}\frac{1}{p_2(a, i)} + \sum_{i=1}^{k}\frac{1}{p_2(b, i)}\right)\right]$$

□

Heuristic Analysis:
Looking at the case of LLM A = LLM B, it is trickier to find a heuristic for producing candidates for C_1, C_2, and $a_1,..., a_k, b_1,..., b_k$ than for est. In particular, there are 2 simultaneously competing effects at play: reducing variance corresponds to increasing $(\Delta z_1 - \Delta z_2)^2$, but reducing Δz_2^2 and Δz_1^2.

Despite it being trickier to produce a theoretically-informed heuristic for $est2$ compared to est, $est2$ has a notable practical advantage: unlike est, $est2$ often works as a stand-alone estimator for determining the ownership identity of LLM B in practice.

To see why, let us recall that $est2$ converges to Δb when LLM A = LLM B. From the definition of Δb, it follows that $\Delta b = 0$ whenever logit biases are 0 or uniformly applied to the tokens $a_1,..., a_k, b_1,..., b_k$. This means that for $\Delta b \neq 0$ to occur, an adversary would have to not only go out of their way to apply heterogeneous logit biases to $a_1,..., a_k, b_1,..., b_k$, but they would

have likely had to guess which tokens we were going to use in creating the estimator $est2$ in the first place (this, of course, precludes cleverer adversaries who automate adding small, random logit biases to every token). Moreover, if $a_1, ..., a_k, b_1, ..., b_k$ are valuable tokens for LLM performance (e.g. numerical), then applying logit biases to them can significantly degrade the LLM's performance. In practice, logit biases are rarely applied, and when they are, they are often uniformly applied. This means that it is usually the case that $\Delta b = 0$. Thus, in many practical scenarios, a simple test for whether LLM A = LLM B is given by checking if $est2$ converges to 0.

By contrast, est estimates temperature T. There is no universal standard for what the default temperature is in practical LLM deployments. In practice, T is often set anywhere from 0.7 to 1 and sometimes out of this range. One can make the counterargument that T is often set to a multiple of 0.1 and therefore est's convergence to a multiple of 0.1 is a sign that LLM A = LLM B. This is valid when this assumption is true, but not when the adversary simply sets the temperature of LLM B to an irregular value.

Naive Method I

For the rest of this "Naive Method I" subsection, we are assuming all assumptions made during its presentation in the "Determining LLM Ownership Identity" section, in addition to the prevailing assumptions being made throughout the entire "Asymptotic Analysis: Independent g_j's" section.

With the assumptions and notation of Naive Method I, we have:

$$\phi(g_1, g_2, g_3, g_4) = \frac{\Delta z_1 - \Delta z_2}{g_1 - g_2} - \frac{\Delta z_3 - \Delta z_4}{g_3 - g_4}$$

Then $var(\phi(g_1, g_2, g_3, g_4)) = var\left(\frac{\Delta z_1 - \Delta z_2}{g_1 - g_2}\right) + var\left(\frac{\Delta z_3 - \Delta z_4}{g_3 - g_4}\right)$ due to the independence

of the g_j's.

We can rewrite this as:

$$var(\phi(g_1, g_2, g_3, g_4)) = var(est(C_1; C_2; a_1, ..., a_k; b_1, ..., b_k)) + var(est(C_3; C_4; a_1', ..., a_{k'}'; b_1', ..., b_{k'}'))$$

Heuristic analysis of $var(\phi(g_1, g_2, g_3, g_4))$ then follows from the "The estimator est" subsection.

Naive Method II

For the rest of this "Naive Method II" subsection, we are assuming all assumptions made during its presentation in the "Determining LLM Ownership Identity" section, in addition to the prevailing assumptions being made throughout the entire "Asymptotic Analysis: Independent g_j's" section.

With the assumptions and notation of Naive Method II, we have:

$$\phi(g_1, g_2, g_3, g_4) = \frac{g_2 \Delta z_1 - g_1 \Delta z_2}{g_1 - g_2} - \frac{g_4 \Delta z_3 - g_3 \Delta z_4}{g_3 - g_4}.$$

Then $var(\phi(g_1, g_2, g_3, g_4)) = var\left(\frac{g_2 \Delta z_1 - g_1 \Delta z_2}{g_1 - g_2}\right) + var\left(\frac{g_4 \Delta z_3 - g_3 \Delta z_4}{g_3 - g_4}\right)$ due to the

independence of the g_j's.

We can rewrite this as:
$$var(\phi(g_1, g_2, g_3, g_4)) = var(est2(C_1; C_2; a_1,..., a_k; b_1,..., b_k)) + var(est2(C_3; C_4; a_1,..., a_k; b_1,..., b_k))$$

Heuristic analysis of $var(\phi(g_1, g_2, g_3, g_4))$ then follows from the "The estimator est2" subsection.

Deferred Proofs (★★)
Definitions:

- Let $X_{j,i} = \dfrac{n_j(a, i)}{N_j}$ for $i = 1 : k_j$

- Let $Y_{j,i} = \dfrac{n_j(b, i)}{N_j}$ for $i = 1 : k_j$

Theorem 9: $var(g_j) \to \dfrac{1}{N_j}\left(\displaystyle\sum_{i=1}^{k_j} \frac{1}{p_j(a, i)} + \sum_{i=1}^{k_j} \frac{1}{p_j(b, i)}\right)$ as $N_j \to \infty$

Proof:
Suppose that $N_j \to \infty$.
First of all, note that we have:

$$g_j(X_{j,1},..., X_{j,k_j}, Y_{j,1},..., Y_{j,k_j}) = \sum_{i=1}^{k_j} ln(N_j X_{j,i}) - \sum_{i=1}^{k_j} ln(N_j Y_{j,i}) = \sum_{i=1}^{k_j} ln(X_{j,i}) - \sum_{i=1}^{k_j} ln(Y_{j,i})$$

Note that $g_j(X_{j,1},..., X_{j,k_j}, Y_{j,1},..., Y_{j,k_j})$'s function is independent of N_j. Furthermore, note that $X_{j,i}$ and $Y_{j,i}$ are consistent estimators of $p_j(a, i)$ and $p_j(b, i)$ respectively. Therefore, by the delta method, we have:
$$g_j(X_{j,1},..., X_{j,k_j}, Y_{j,1},..., Y_{j,k_j}) \approx g_j(p_j(a, 1),..., p_j(a, k_j), p_j(b, 1),..., p_j(b, k_j))$$

$$+\sum_{i=1}^{k_j} \frac{\partial g}{\partial x_{j,i}}(p_j(a,1),...,p_j(a,k_j),p_j(b,1),...,p_j(b,k_j))(X_{j,i}-p_j(a,i))$$

$$+\sum_{i=1}^{k_j} \frac{\partial g}{\partial y_{j,i}}(p_j(a,1),...,p_j(a,k_j),p_j(b,1),...,p_j(b,k_j))(Y_{j,i}-p_j(b,i)) \quad (4)$$

Let $A_i = \dfrac{\partial g}{\partial x_{j,i}}(p_j(a,1),...,p_j(a,k_j),p_j(b,1),...,p_j(b,k_j))$

Let $B_i = \dfrac{\partial g}{\partial y_{j,i}}(p_j(a,1),...,p_j(a,k_j),p_j(b,1),...,p_j(b,k_j))$

Then we have:

$$var(g_j(X_{j,1},...,X_{j,k_j},Y_{j,1},...,Y_{j,k_j}))$$

$$\approx var\left(\sum_{i=1}^{k_j} A_i(X_{j,i}-p_j(a,i)) + \sum_{i=1}^{k_j} B_i(Y_{j,i}-p_j(b,i))\right)$$

$$= \sum_{i=1}^{k_j} A_i^2 var(X_{j,i}-p_j(a,i)) + \sum_{i=1}^{k_j} B_i^2 var(Y_{j,i}-p_j(b,i))$$

$$+2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} A_i A_l Cov(X_{j,i}-p_j(a,i), X_{j,l}-p_j(a,l))$$

$$+2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} B_i B_l Cov(Y_{j,i}-p_j(b,i), Y_{j,l}-p_j(b,l))$$

$$+2\sum_{i=1}^{k_j}\sum_{l=1}^{k_j} A_i B_l Cov(X_{j,i}-p_j(a,i), Y_{j,l}-p_j(b,l))$$

$$= \sum_{i=1}^{k_j} A_i^2 var(X_{j,i}) + \sum_{i=1}^{k_j} B_i^2 var(Y_{j,i})$$

$$+2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} A_i A_l Cov(X_{j,i}, X_{j,l})$$

$$+2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} B_i B_l Cov(Y_{j,i}, Y_{j,l})$$

$$+2\sum_{i=1}^{k_j}\sum_{l=1}^{k_j} A_i B_l Cov(X_{j,i}, Y_{j,l})$$

$$= \sum_{i=1}^{k_j} A_i^2 var\left(\frac{n_j(a,i)}{N_j}\right) + \sum_{i=1}^{k_j} B_i^2 var\left(\frac{n_j(b,i)}{N_j}\right)$$

$$+2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} A_i A_l Cov\left(\frac{n_j(a,i)}{N_j}, \frac{n_j(a,l)}{N_j}\right)$$

$$+2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} B_i B_l Cov\left(\frac{n_j(b,i)}{N_j}, \frac{n_j(b,l)}{N_j}\right)$$

$$+2\sum_{i=1}^{k_j}\sum_{l=1}^{k_j} A_i B_l Cov\left(\frac{n_j(a,i)}{N_j}, \frac{n_j(b,l)}{N_j}\right)$$

Due to (*) preventing overlapping generations among $a_{1,j},...,a_{k_j,j}, b_{1,j},...,b_{k_j,j}$ for each j, the $n_j(a,i)$'s and $n_j(b,i)$'s are components of a multinomial distribution for each j. The following are facts from multinomial distributions:

- $var(n_j(a,i)) = N_j p_j(a,i)(1 - p_j(a,i))$
- $var(n_j(b,i)) = N_j p_j(b,i)(1 - p_j(b,i))$
- $Cov(n_j(a,i), n_j(a,l)) = -N_j p_j(a,i)p_j(a,l)$
- $Cov(n_j(b,i), n_j(b,l)) = -N_j p_j(b,i)p_j(b,l)$
- $Cov(n_j(a,i), n_j(b,l)) = -N_j p_j(a,i)p_j(b,l)$

So the above expression becomes:

$$\sum_{i=1}^{k_j} \frac{A_i^2}{N_j^2} N_j p_j(a,i)(1-p_j(a,i)) + \sum_{i=1}^{k_j} \frac{B_i^2}{N_j^2} N_j p_j(b,i)(1-p_j(b,i))$$

$$+2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} \frac{A_i A_l}{N_j^2}(-N_j p_j(a,i)p_j(a,l))$$

$$+2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} \frac{B_i B_l}{N_j^2}(-N_j p_j(b,i)p_j(b,l))$$

$$+2\sum_{i=1}^{k_j}\sum_{l=1}^{k_j} \frac{A_i B_l}{N_j^2}(-N_j p_j(a,i)p_j(b,l))$$

$$= \frac{1}{N_j}\left(\sum_{i=1}^{k_j} A_i^2 p_j(a,i)(1-p_j(a,i)) + \sum_{i=1}^{k_j} B_i^2 p_j(b,i)(1-p_j(b,i))\right)$$

$$-2\sum_{1\leqslant i<l\leqslant k_j}^{k_j} A_i A_l(p_j(a,i)p_j(a,l))$$

$$-2 \sum_{1 \leqslant i < l \leqslant k_j}^{k_j} B_i B_l (p_j(b, i) p_j(b, l))$$

$$-2 \sum_{i=1}^{k_j} \sum_{l=1}^{k_j} A_i B_l (p_j(a, i) p_j(b, l)))$$

We now compute A_i and B_i.

$$g_j(x_{j,1}, ..., x_{j,k_j}, y_{j,1}, ..., y_{j,k_j}) = \sum_{i=1}^{k_j} ln(X_{j,i}) - \sum_{i=1}^{k_j} ln(Y_{j,i})$$

So $\dfrac{\partial g}{\partial x_{j,i}}(x_{j,1}, ..., x_{j,k_j}, y_{j,1}, ..., y_{j,k_j}) = \dfrac{1}{x_{j,i}}$ and $\dfrac{\partial g}{\partial y_{j,i}}(x_{j,1}, ..., x_{j,k_j}, y_{j,1}, ..., y_{j,k_j}) = -\dfrac{1}{y_{j,i}}$

Then we have:

$$A_i = \frac{\partial g}{\partial x_{j,i}}(p_j(a, 1), ..., p_j(a, k_j), p_j(b, 1), ..., p_j(b, k_j))$$

$$= \frac{1}{p_j(a, i)}$$

And similarly:

$$B_i = \frac{\partial g}{\partial y_{j,i}}(p_j(a, 1), ..., p_j(a, k_j), p_j(b, 1), ..., p_j(b, k_j))$$

$$= -\frac{1}{p_j(b, i)}$$

Therefore, the above expression becomes:

$$\frac{1}{N_j}(\sum_{i=1}^{k_j} \frac{1}{p_j(a, i)^2} p_j(a, i)(1 - p_j(a, i)) + \sum_{i=1}^{k_j} \left(-\frac{1}{p_j(b, i)}\right)^2 p_j(b, i)(1 - p_j(b, i))$$

$$-2 \sum_{1 \leqslant i < l \leqslant k_j}^{k_j} \frac{1}{p_j(a, i)} \frac{1}{p_j(a, l)}(p_j(a, i) p_j(a, l))$$

$$-2 \sum_{1 \leqslant i < l \leqslant k_j}^{k_j} \left(-\frac{1}{p_j(b, i)}\right)\left(-\frac{1}{p_j(b, l)}\right)(p_j(b, i) p_j(b, l))$$

$$-2 \sum_{i=1}^{k_j} \sum_{l=1}^{k_j} \frac{1}{p_j(a, i)}\left(-\frac{1}{p_j(b, l)}\right)(p_j(a, i) p_j(b, l)))$$

$$= \frac{1}{N_j} \left(\sum_{i=1}^{k_j} \left[\frac{1}{p_j(a, i)} - 1 \right] + \sum_{i=1}^{k_j} \left[\frac{1}{p_j(b, i)} - 1 \right] \right.$$

$$-2 \sum_{1 \leqslant i < l \leqslant k_j}^{k_j} 1$$

$$-2 \sum_{1 \leqslant i < l \leqslant k_j}^{k_j} 1$$

$$\left. +2 \sum_{i=1}^{k_j} \sum_{l=1}^{k_j} 1 \right)$$

$$= \frac{1}{N_j} \left(\sum_{i=1}^{k_j} \frac{1}{p_j(a, i)} + \sum_{i=1}^{k_j} \frac{1}{p_j(b, i)} \right)$$

$\square$

Theorem 10: Each g_j is asymptotically a normal random variable as $N_j \to \infty$.

Proof:

Suppose $N_j \to \infty$.

From equation (4), we have the following:

$$g_j(X_{j,1}, ..., X_{j,k_j}, Y_{j,1}, ..., Y_{j,k_j}) \approx c + \sum_{i=1}^{k_j} d_i(X_{j,i} - p_j(a, i)) + \sum_{i=1}^{k_j} e_i(Y_{j,i} - p_j(b, i))$$

For some constants c, d_i, e_i. Since normality is preserved under an additive constant, it suffices to show that the following is normal:

$$\sum_{i=1}^{k_j} d_i X_{j,i} + \sum_{i=1}^{k_j} e_i Y_{j,i}$$

$$= \sum_{i=1}^{k_j} d_i \frac{n_j(a, i)}{N_j} + \sum_{i=1}^{k_j} e_i \frac{n_j(b, i)}{N_j}$$

Write $n_j(a, i)$ as a sum of IID indicator random variables $I_j(a, i, t)$ for $t = 1 : N_j$, each indicator random variable indicating whether a_i was a prefix string of the tth generated output of LLM B on context C_j. Define $I_j(b, i, t)$ similarly.

Thus, we have:

$$\sum_{i=1}^{k_j} d_i \frac{n_j(a, i)}{N_j} + \sum_{i=1}^{k_j} e_i \frac{n_j(b, i)}{N_j}$$

$$= \sum_{i=1}^{k_j} d_i \frac{\sum_{t=1}^{N_j} I_j(a, i, t)}{N_j} + \sum_{i=1}^{k_j} e_i \frac{\sum_{t=1}^{N_j} I_j(b, i, t)}{N_j}$$

$$= \frac{1}{N_j} \sum_{t=1}^{N_j} \left(\sum_{i=1}^{k_j} d_i I_j(a, i, t) + \sum_{i=1}^{k_j} e_i I_j(b, i, t) \right)$$

Take $Z_t = \sum_{i=1}^{k_j} d_i I_j(a, i, t) + \sum_{i=1}^{k_j} e_i I_j(b, i, t)$. Note that the Z_t's are IID.

The expression then becomes $\frac{1}{N_j} \sum_{t=1}^{N_j} Z_t$, which is asymptotically normal by the central limit theorem. □

Lemma 4: $Bias(g_j) = E[g_j] - c_j = O(1/N_j)$

Proof:

Recall from the proof of theorem 9 that we may write $g_j = \sum_{i=1}^{k_j} ln(X_{j,i}) - \sum_{i=1}^{k_j} ln(Y_{j,i})$.

Then:

$$E[g_j] - c_j = \sum_{i=1}^{k_j} E[ln(X_{j,i})] - \sum_{i=1}^{k_j} E[ln(Y_{j,i})] - \left(\sum_{i=1}^{k_j} ln(p_j(a, i)) - \sum_{i=1}^{k_j} ln(p_j(b, i)) \right)$$

$$= \sum_{i=1}^{k_j} (E[ln(X_{j,i})] - ln(p_j(a, i))) - \sum_{i=1}^{k_j} (E[ln(Y_{j,i})] - ln(p_j(b, i)))$$

Since k_j's are fixed, it suffices to show that each $E[ln(X_{j,i})] - ln(p_j(a, i))$ and each $E[ln(Y_{j,i})] - ln(p_j(b, i))$ term is $O(1/N_j)$. We will show this for $E[ln(X_{j,i})] - ln(p_j(a, i))$. The proof for the $E[ln(Y_{j,i})] - ln(p_j(b, i))$ case is identical.

We have by the 2nd-order Taylor approximation that:

$$ln(X_{j,i}) = ln(p_j(a, i)) + \frac{1}{p_j(a, i)}(X_{j,i} - p_j(a, i)) - \frac{1}{2p_j(a, i)^2}(X_{j,i} - p_j(a, i))^2 + R$$ where R is

the remainder term.

Thus, we have:

$$E[ln(X_{j,i})] - ln(p_j(a, i)) = \frac{1}{p_j(a, i)} E[(X_{j,i} - p_j(a, i))] - \frac{1}{2p_j(a, i)^2} E\left[(X_{j,i} - p_j(a, i))^2 \right] + E[R]$$

Here are some standard facts from binomial distributions:

- $E\left[(X_{j,i} - p_j(a, i))^2 \right] = O(1/N_j)$
- $|X_{j,i} - p_j(a, i)| = O_p\left(1/N_k^{1/2} \right)$

Note that the $\dfrac{1}{p_j(a, i)} E[(X_{j,i} - p_j(a, i))]$ term is simply 0.

Furthermore, the $-\dfrac{1}{2p_j(a, i)^2} E\left[(X_{j,i} - p_j(a, i))^2\right]$ term is $O(1/N_j)$.

$|R| = O\left(|X_{j,i} - p_j(a, i)|^3\right)$ follows from a standard bound for Taylor polynomial remainders.

Thus, $|E[R]| \leqslant E[|R|] = O\left(E\left[|X_{j,i} - p_j(a, i)|^3\right]\right)$.

Then

$|X_{j,i} - p_j(a, i)|^3 = O_p\left(1/N_k^{3/2}\right) \implies O\left(E\left[|X_{j,i} - p_j(a, i)|^3\right]\right) = O\left(1/N_k^{3/2}\right) = O(1/N_k)$.

Therefore, $|E[R]| = O(1/N_k)$.

Therefore, $E[ln(X_{j,i})] - ln(p_j(a, i)) = O(1/N_k)$.

This completes the proof. $\square$

Theorem 14: $Bias(\phi(g_1,..., g_m)) = E[\phi(g_1,..., g_m)] - \phi(c_1,..., c_m) = O\left(\displaystyle\sum_{j=1}^{m} 1/N_j\right)$

Proof:

Let $g = (g_1,..., g_m)^T$ and $c = (c_1,..., c_m)^T$. By the 2nd-order Taylor approximation, we have that:

$$\phi(g) = \phi(c) + \nabla\phi(c)^T(g - c) + \frac{1}{2}(g - c)^T H_\phi(c)(g - c)^T + R.$$

Thus, we have:

$$E[\phi(g)] - \phi(c) = \nabla\phi(c)^T E[g - c] + E\left[\frac{1}{2}(g - c)^T H_\phi(c)(g - c)^T\right] + E[R]$$

We have $\nabla\phi(c)^T E[g - c] = O\left(\displaystyle\sum_{j=1}^{m} 1/N_j\right)$ by lemma 4.

We have:

$$|E\left[\frac{1}{2}(g - c)^T H_\phi(c)(g - c)^T\right]| \leqslant E\left[|\frac{1}{2}(g - c)^T H_\phi(c)(g - c)|\right]$$

$$\leqslant E\left[\frac{1}{2}||H_\phi(c)||_2||g - c||_2^2\right]$$

$$= \frac{1}{2}||H_\phi(c)||_2 E\left[||g - c||_2^2\right]$$

$$= const \cdot \sum_{j=1}^{m} E\left[(g_j - c_j)^2\right]$$

We then have:

$$E\left[(g_j - c_j)^2\right] = var(g_j - c_j) + E[g_j - c_j]^2$$

$= var(g_j) + O(1/N_j)^2$ (by lemma 4)

$= O(1/N_j) + O(1/N_j)^2$ (by theorem 9)

$= O(1/N_j)$

Thus, $|E\left[\frac{1}{2}(g-c)^T H_\phi(c)(g-c)^T\right]| = O\left(\sum_{j=1}^{m} 1/N_j\right)$.

$|R| = O\left(||g-c||^3\right)$ follows from a standard bound for Taylor polynomial remainders.

Then:

$|E[R]| \leqslant E[|R|]$

$= E\left[O\left(||g-c||^3\right)\right]$

$\leqslant E\left[O\left(\left(\sum_{j=1}^{m} |g_j - c_j|\right)^3\right)\right]$

Claim that we'll prove at the end of this proof: $|g_j - c_j| = O_p\left(1/\sqrt{N_j}\right)$

Then:

$$E\left[O\left(\left(\sum_{j=1}^{m} |g_j - c_j|\right)^3\right)\right] = O\left(\left(\sum_{j=1}^{m} 1/\sqrt{N_j}\right)^3\right)$$

$$= O\left(\left(1/\sqrt{N_{min}}\right)^3\right)$$

$$= O\left(1/N_{min}^{-3/2}\right)$$

$$= O\left(\sum_{j=1}^{m} 1/N_j^{-3/2}\right)$$

$$= O\left(\sum_{j=1}^{m} 1/N_j\right)$$

Thus, $|E[R]| = O\left(\sum_{j=1}^{m} 1/N_j\right)$. We conclude that $E[\phi(g)] - \phi(c) = O\left(\sum_{j=1}^{m} 1/N_j\right)$.

It now remains to prove $|g_j - c_j| = O_p\left(1/\sqrt{N_j}\right)$.

By bias-variance decomposition, we have:

$E\left[(g_j - c_j)^2\right] = var(g_j) + (E[g_j] - c_j)^2$

$= O(1/N_j)$ by theorem 9 and lemma 4.

This implies that $E\left[N_j(g_j - c_j)^2\right] = O(1)$.

Markov's inequality gives us:

$$P\left(|g_j - c_j| > \frac{M}{\sqrt{N}}\right) = P\left(N_j(g_j - c_j)^2 > M^2\right) \leqslant \frac{E\left[N_j(g_j - c_j)^2\right]}{M^2} \leqslant \frac{const}{M^2}.$$

This implies $|g_j - c_j| = O_p\left(1/\sqrt{N_j}\right)$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

Asymptotic Analysis: Non-independent g_j's (The General Case)

> - Throughout this section, $\phi(x_1,...,x_m)$ will be an arbitrary smooth function unless otherwise stated.
> - Throughout this section, we will be assuming †† instead of † (which is a special case of ††; †† is a relaxation of †).

The "Asymptotic Analysis: Independent g_j's" section relied on the assumption that the g_j's were decoupled and independent. In other words, each g_j was assumed to be constructed from an independent, self-contained experiment involving running LLM B many times on context C_j and recording number of occurrences of $a_{1,j},..., a_{k_j,j}, b_{1,j},..., b_{k_j,j}$. The "What is an experiment?" subsection provided an illustrative example for a scenario where different g_j's can become coupled and non-independent. We consider that situation in this section.

Asymptotic analysis of $\phi(g_1,..., g_m)$ in the setting of coupled g_j's runs into more complications compared to the independent setting and will require careful additional machinery, which will be left to future work. For now, we will present the bare minimum core theorems for $\phi(g_1,..., g_m)$'s asymptotic properties in the non-independent setting that are necessary for proving the general validity of the estimator algorithm and hypothesis test.

Fortunately, many analogous theorems can be proven in the non-independent case that mirror the structure of the theorems in the independent case. We present these theorems in the language of *experiments*, which was first introduced in "What is an experiment?"

The Core Theorems (General Case)

Theorem 16: $\phi(g_1,..., g_m)$ is asymptotically a normal random variable as $N^{(1)},..., N^{(w)} \to \infty$.

Proof:

First, note that by the law of large numbers, for any $n_{l,r} \in O_r$, $\dfrac{n_{l,r}}{N^{(r)}} \to p_{l,r}$ for some

probability $p_{l,r}$ as $N^{(r)} \to \infty$. Define $X_l^{(r)} = \dfrac{n_{l,r}}{N^{(r)}}$.

Thus far, we've been treating g_j's as the atomic building blocks of $\phi(g_1,...,g_m)$, but there is an object that is even more granular: $n_j(a, i)$'s and $n_j(b, i)$'s. Recall the definition that

$$g_j = ln\left(\prod_{i=1}^{k_j} \frac{n_j(a, i)}{n_j(b, i)}\right) = \sum_{i=1}^{k_j} ln(n_j(a, i)) - \sum_{i=1}^{k_j} ln(n_j(b, i)).$$ Thus, we may view $\phi(g_1,...,g_m)$ as

a function of $n_j(a, i)$'s and $n_j(b, i)$'s instead of as a function of g_j's. Equivalently, we may view $\phi(g_1,...,g_m)$ as a function of $n_{l,r}$'s. Equivalently, we may view $\phi(g_1,...,g_m)$ as a function of $X_l^{(r)}$'s.

We wish to apply the delta method on $\phi(g_1,...,g_m)$ as a function of $X_l^{(r)}$'s, but we must first ensure that when $\phi(g_1,...,g_m)$ is expressed as a function of $X_l^{(r)}$'s, it is a fixed function that does not depend on the $N^{(r)}$'s.

For every j, we know that $n_j(a, i)$ and $n_j(b, i)$ for $i = 1 : k_j$ all belong to the same O_r. Fix r.

Consider again the equation $g_j = \sum_{i=1}^{k_j} ln(n_j(a, i)) - ln(n_j(b, i))$. Let us consider index i in this

summation. Refer to $n_j(a, i)$ and $n_j(b, i)$ as $n_{l_1,r}$ and $n_{l_2,r}$ respectively. Then

$ln(n_j(a, i)) - ln(n_j(b, i)) = ln\left(N^{(r)} X_{l_1}^{(r)}\right) - ln\left(N^{(r)} X_{l_2}^{(r)}\right) = ln\left(X_{l_1}^{(r)}\right) - ln\left(X_{l_2}^{(r)}\right).$ So we see

that each term in the summation of g_j does not depend on $N^{(r)}$. Therefore, g_j is a fixed function of $X_l^{(r)}$ terms. Thus, $\phi(g_1,...,g_m)$ is a fixed function of $X_l^{(r)}$ terms.

We may now apply the delta method:

$$\phi(g_1,...,g_m) \approx c + \sum_{r=1}^{w} \sum_{l=1}^{k^{(r)}} d_{l,r}\left(X_l^{(r)} - p_{l,r}\right)$$ as $N^{(1)},..., N^{(w)} \to \infty$ for some fixed constants c

and $d_{l,r}, r = 1 : w, l = 1 : k^{(r)}$.

Note that each $\sum_{l=1}^{k^{(r)}} d_{l,r} X_l^{(r)}$ is independent for $r = 1 : w$ due to the O_r's being statistically

independent. Therefore, to prove asymptotic normality of $\phi(g_1,...,g_m)$, it suffices to then

prove that $\sum_{l=1}^{k^{(r)}} d_{l,r} X_l^{(r)}$ is asymptotically normal.

We have:

$$\sum_{l=1}^{k^{(r)}} d_{l,r} X_l^{(r)} = \sum_{l=1}^{k^{(r)}} d_{l,r} \frac{n_{l,r}}{N^{(r)}}$$

Write each $n_{l,r}$ as a sum of IID indicator random variables $I_r(l,t)$ for $t = 1 : N^{(r)}$, which record whether or not the event that $n_{l,r}$ is tracking occurred on trial/run t of experiment r. We now have:

$$\sum_{l=1}^{k^{(r)}} d_{l,r} \frac{n_{l,r}}{N^{(r)}} = \sum_{l=1}^{k^{(r)}} d_{l,r} \frac{\sum_{t=1}^{N^{(r)}} I_r(l,t)}{N^{(r)}}$$

$$= \frac{1}{N^{(r)}} \sum_{t=1}^{N^{(r)}} \sum_{l=1}^{k^{(r)}} d_{l,r} I_r(l,t)$$

Take $Z_t^{(r)} = \sum_{l=1}^{k^{(r)}} d_{l,r} I_r(l,t)$. Note that the $Z_t^{(r)}$'s are IID.

The expression then becomes $\frac{1}{N^{(r)}} \sum_{t=1}^{N^{(r)}} Z_t^{(r)}$, which is asymptotically normal by the central limit theorem. This completes the proof. $\square$

Theorem 17: $var(\phi(g_1,...,g_m))$ is $\Theta\left(\sum_{r=1}^{w} \frac{1}{N^{(r)}}\right)$ as $N^{(1)},...,N^{(w)} \to \infty$

Proof:
From theorem 16's proof, we see that as $N^{(1)},...,N^{(w)} \to \infty$, $\phi(g_1,...,g_m)$ can be written, up to an additive constant, as $\sum_{r=1}^{w} \sum_{l=1}^{k^{(r)}} d_{l,r} X_l^{(r)} = \sum_{r=1}^{w} \left(\frac{1}{N^{(r)}} \sum_{t=1}^{N^{(r)}} Z_t^{(r)}\right)$.

Since different O_r's are statistically independent, we have:

$$var(\phi(g_1,...,g_m)) = var\left(\sum_{r=1}^{w} \sum_{l=1}^{k^{(r)}} d_{l,r} X_l^{(r)}\right) = var\left(\sum_{r=1}^{w} \left(\frac{1}{N^{(r)}} \sum_{t=1}^{N^{(r)}} Z_t^{(r)}\right)\right) = \sum_{r=1}^{w} var\left(\frac{1}{N^{(r)}} \sum_{t=1}^{N^{(r)}} Z_t^{(r)}\right)$$

Since the $Z_t^{(r)}$'s are IID for any given r, we can give $Z_t^{(r)}, t = 1 : N^{(r)}$ the common variance $var^{(r)}$ and write:

$$\sum_{r=1}^{w} var\left(\frac{1}{N^{(r)}} \sum_{t=1}^{N^{(r)}} Z_t^{(r)}\right) = \sum_{r=1}^{w} \frac{1}{N^{(r)}} var^{(r)} = \Theta\left(\sum_{r=1}^{w} \frac{1}{N^{(r)}}\right) \qquad \square$$

Comment: Since we know $var(\phi(g_1,...,g_m))$ is $O\left(\sum_{r=1}^{w}\dfrac{1}{N^{(r)}}\right)$ as $N^{(1)},...,N^{(w)} \to \infty$, then taking the $N^{(r)}$'s to be fixed proportions of N (the total number of samples/calls/runs across all experiments) results in $O(1/N)$ variance decay for $\phi(g_1,...,g_m)$. This matches the best possible parametric decay rate $O(1/N)$ for the variance of an estimator satisfying standard regularity conditions.

Comment 2: $N = N^{(1)} + ... + N^{(w)}$ is the number of calls to LLM B in the generic, non-independent g_j setting. In general, $N_1 + ... + N_m > N$ because the sum double-counts calls, since multiple N_j's may come from the exact same experiment. $N = N_1 + ... + N_m$ is only true if we assume each g_j was created from a self-contained, independent experiment (i.e. what was assumed in the previous section).

Theorem 18: $\phi(g_1,...,g_m)$ is a consistent estimator of $\phi(c_1,...,c_m)$ as $N^{(1)},...,N^{(w)} \to \infty$.

Proof:
This is a restatement of Lemma 2. Note that at no intermediate step of Lemma 2's proof was the assumption of g_j independence ever used, making it valid in this section as well. □

Theorem 19: $Bias(\phi(g_1,...,g_m)) = E[\phi(g_1,...,g_m)] - \phi(c_1,...,c_m) = O\left(\sum_{r=1}^{w}1/N^{(r)}\right)$

Proof:

Theorem 14 from the previous section tells us that $Bias(\phi(g_1,...,g_m)) = O\left(\sum_{j=1}^{m}1/N_j\right)$. Note that we are allowed to apply theorem 14 in this setting since the proof of theorem 14 did not require assuming the g_j's are independent.

We then have $O\left(\sum_{j=1}^{m}1/N_j\right) = O\left(m\sum_{r=1}^{w}1/N^{(r)}\right) = O\left(\sum_{r=1}^{w}1/N^{(r)}\right)$, since m is fixed. □

Comment: From theorem 16, we have that $\sigma(\phi(g_1,...,g_m))$ decays at rate $\Theta\left(\sqrt{\sum_{r=1}^{w}1/N^{(r)}}\right)$, which asymptotically dominates $Bias(\phi(g_1,...,g_m)) = O\left(\sum_{r=1}^{w}1/N^{(r)}\right)$. In other words, $Bias(\phi(g_1,...,g_m)) = o(\sigma(\phi(g_1,...,g_m))$. Combined with the fact that

$\phi(g_1,...,g_m)$ is asymptotically normal, we may treat bias as negligible ($= 0$) as $N^{(1)},...,N^{(w)} \to \infty$ in many analyses.

Conclusion
TODO

References
TODO

Appendix

Enforcing $\dagger, 2$**:**
The math behind the theorems in our "Theoretical Consequences of LLM A = LLM B" section hinges on $\dagger, 2$. In other words, we required that given any generated output string from LLM A, $t \in \{a_1,...,a_k, b_1,...,b_k\}$ is a prefix of that string iff it is the first generated token of that string. $\dagger, 2$ is what makes the following 2 sets of definitions equivalent when LLM A = LLM B:

Version 1:
- Let $p_j(a, i)$ be the true probability of a_i being a prefix string of a generated output from LLM B on context C_j (unknown to us)
- Let $p_j(b, i)$ be the true probability of b_i being a prefix string of a generated output from LLM B on context C_j (unknown to us)
- Let $n_j(a, i)$ be the actual number of times a_i appeared as a prefix string of a generated output from LLM B on context C_j (known to us)
- Let $n_j(b, i)$ be the actual number of times b_i appeared as a prefix string of a generated output from LLM B on context C_j (known to us)

Version 2:
- Let $p_j(a, i)$ be the true probability of a_i in the first-token distribution of LLM B with respect to context C_j (unknown to us)
- Let $p_j(b, i)$ be the true probability of b_i in the first-token distribution of LLM B with respect to context C_j (unknown to us)
- Let $n_j(a, i)$ be the actual number of times a_i appeared as the first token of LLM B on context C_j (known to us)
- Let $n_j(b, i)$ be the actual number of times b_i appeared as the first token of LLM B on context C_j (known to us)

As an easy-to-understand motivating example, consider this situation: Suppose tokens '1', '2', and '12' are all in the vocabulary of LLM A . When the LLM returns string "12", we don't know

if it returned '1' + '2' or '12'. What should we do to prevent this situation?

We will now discuss some practical ways to prevent this ambiguity. The first method works even when we don't have access to LLM A's vocabulary list.

Method #1: Forcing the LLM to output only 1 token
LLM deployments often expose a max_tokens parameter, putting a hard cap on the maximum number of tokens the LLM can generate before termination. The easiest way to prevent first token ambiguity is to set max_tokens = 1.

In addition, many consumer-facing LLM providers often stream their output one at a time in string chunks that correspond to individual tokens. From this, we can, in principle, simulate max_tokens = 1 by extracting the string representation of only the first token generated in the LLM's output string.

Method #2: Atomic Tokens
Choosing a_i's and b_i's to be "atomic tokens" of the LLM is a non-"hacky" method to ensure that a_i/b_i is the prefix of an output iff it is the first token of that output. Here, we define "atomic tokens" as tokens whose string representations are not a prefix of any other token's string representation and also do not contain any other token's string representation as a prefix of itself.

If we choose the a_i's and b_i's to be atomic tokens of the LLM, then whenever we see one of them appear as a prefix string of a generated output of the LLM, then we know with certainty that it is the first token of that string with zero ambiguity.

Since we have white-box access to LLM A (and therefore its vocabulary), we can in principle automate the process to find atomic tokens of LLM A.

Many LLMs share atomic tokens in common. Rare, standalone Unicode characters (especially a symbol or emoji) are often atomic tokens. For example:
- Certain emojis: ⭐ (U+2B50), ❇ (U+2747), ☂ (U+2602), 🦊 (U+1F98A), etc.
- Certain rare characters: ⬣ (U+2B23), ❉ (U+2749), ⯆ (U+2BC6), 🜏 (U+1F70F), etc.