Perplexity Cannot Always Tell Right from Wrong

Anonymous Authors[1]

Abstract

Perplexity—a function measuring a model's overall level of "surprise" when encountering a particular output—has gained significant traction in recent years, both as a loss function and as a simple-to-compute metric of model quality. Prior studies have pointed out several limitations of perplexity, often from an empirical manner. Here we leverage recent results on Transformer continuity to show in a rigorous manner how perplexity may be an unsuitable metric for model selection. Specifically, we prove that, if there is *any* sequence that a compact decoder-only Transformer model predicts accurately and confidently—a necessary pre-requisite for strong generalisation—it must imply existence of another sequence with very low perplexity, but not predicted correctly by that same model. Further, by analytically studying iso-perplexity plots, we find that perplexity will not always select for the more accurate model—rather, any increase in model confidence must be accompanied by a commensurate rise in accuracy for the new model to be selected.

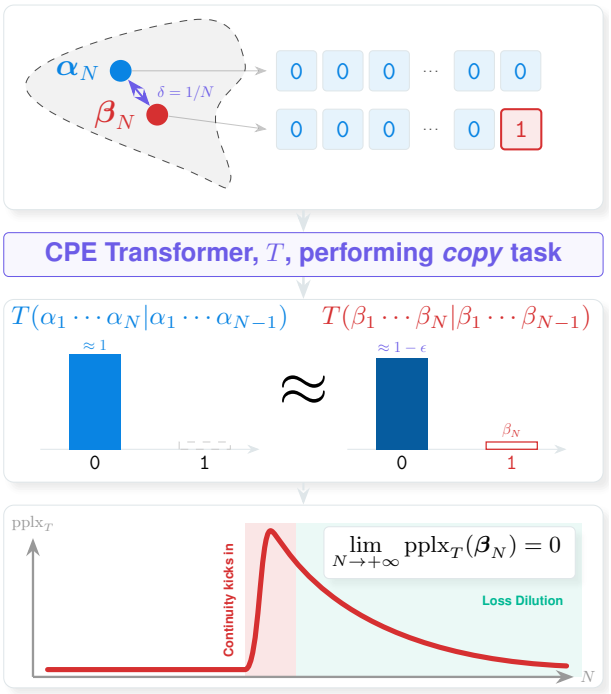


Figure 1. Using the continuity result of Pasten et al. (2025), we show that, if a (compact) Transformer T is confident in copying *any* long enough sequence $\boldsymbol{\alpha}_N$, then there must exist $\boldsymbol{\beta}_N$ which T fails to copy, yet, log-perplexity will tend to zero as N grows.

1. Introduction

Perplexity is a measure of a model's "surprise" when observing ground-truth data; assuming a model's output distribution, Q, over a space of classes, $\mathcal{C}$, used to approximate a ground-truth distribution, P, it can be expressed as $\exp \sum_{k \in \mathcal{C}} -p_k \log q_k$, and, since it is easy to compute over any classification task (including tokenised data), it has become a popular function for evaluating sequential machine learning models when no other performance metric is readily computable. However, even though it is simple to use and interpret, we provide novel evidence that perplexity should not be blindly trusted as a model selection objective.

This is a result that has been informally observed in several venues (Fang et al. (2025); Hu et al. (2024); Hsieh et al. (2024)). Intuitively, it stems from the fact that perplexity encodes both confidence in prediction as well as correctness. Subsequently, a model with a lower accuracy (e.g. more answer tokens predicted incorrectly in the context of language models), but with better-calibrated confidence, may have lower perplexity—and thus be preferred.

Making use of recent theoretical developments (Pasten et al., 2025), in this work we (1) **formalise** these observations, (2) prove they **must** occur when using contemporary decoder-only Transformers, and (3) demonstrate **how** to elicit them reliably. We also provide an analytic account of how perplexity drives unfavourable "offsets" between confident and performant models in perplexity's decision space. Our results indicate that **any** confident model will necessarily introduce inputs which it will likely get *wrong*, but at a *negligible effect* to perplexity (see Figure 1). Note that confident

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

models are *necessary* for reliable long-range predictions in language models, meaning that our results will apply to any competent contemporary language model.

Specifically, our work contributes the following:

1. We prove that, for a wide class of decoder-only Transformer-based language models, should the model be highly confident and correct on *any* sufficiently long input sequence, this must imply existence of *another* input where the model's prediction is *wrong*, yet the log-perplexity of that prediction approaches *zero*.
2. We empirically validate this observation by studying *bitstring copy tasks*, both for a custom trained decoder-only Transformer over a small vocabulary, and the Gemma 3 4B large language model (Team et al., 2025).
3. Under certain assumptions on homogeneity of confidence, we study *iso-perplexity curves* in the confidence/accuracy space. These curves reveal "unfavourable regions" where a model grows too confident to justify its own accuracy improvement, and would not be selected against many weaker models. We provide empirical findings that match this analytic observation.

2. Related work

While perplexity (or similarly log-likelihood) is the standard metric for evaluating language models, it has long been known in the broader generative model literature that likelihood does not necessarily correlate with sample quality. For instance, Theis et al. (2015) showed that good likelihood scores can be achieved by models that generate poor samples, and conversely, high-quality generators can yield poor likelihoods. Nalisnick et al. (2019) showed that VAEs or flow-based models can assign higher likelihood to images that are *outside* the training distribution and that therefore do not represent the training data.

In the context of language models, Holtzman et al. (2019) showed a similar disconnect between optimising for likelihood and the generation of high-quality samples. In particular, they show that decoding using Nucleus Sampling leads to better generations (with lower likelihoods) compared to likelihood-maximising approaches such as beam search.

Failures of Perplexity in Long-Context Fang et al. (2025) argue that using perplexity as a metric in long-context is often misleading because useful signal may vanish when averaging perplexity over thousands of tokens. Their work champions the view that the *aggregation* method is the culprit. Our work rigorously proves results related to this observation, while also extending to claim that there is a detrimental, asymmetric relationship between accuracy and model confidence, which complicates the story further.

We highlight that this has been alluded to by other work. Gelberg et al. (2025, Figure 5) showed that models can maintain low perplexity even when relevant information is strictly unreachable, which seems to explain the effectiveness of the popular context extension method YaRN (Peng et al., 2023). Similarly, Liu et al. (2024) and Hsieh et al. (2024) have shown that models often fail to retrieve information 'lost in the middle' of a prompt, despite achieving low overall perplexity scores on those same documents. These findings suggest that perplexity is not necessarily aligned with model performance, especially in long-context regimes.

Exposure Bias Many of the long-range issues we explore will inherently occur once the model is required to *generate* a lot of tokens, which are then reused for its own conditioning. The eventual mismatch between training and generated data is well-documented under the concept of *exposure bias* (Ranzato et al., 2016), which attracted significant attention (e.g., Schmidt (2019); Wang & Sennrich (2020).

Confidence and Calibration A core component of our analysis is the role of model confidence. Guo et al. (2017) famously showed that modern neural networks tend to be miscalibrated and overconfident. In the LLM era, while some argue models are generally calibrated (Kadavath et al., 2022), the incentive to minimise perplexity encourages 'confident' predictions *in-distribution*. Our analysis studies how these training dynamics allow models to trade accuracy for confidence, creating 'unfavourable regions' where a 'confident but wrong model' achieves a better perplexity score than a hesitant but more accurate one.

Theoretical Results on Transformers Our work relies on recent theoretical works regarding limitations of the Transformer architecture. Barbero et al. (2024) identified the phenomenon of representation collapse in decoder-only Transformers. Extending this, Pasten et al. (2025) proved the existence of a 'concentration' of infinite sequence collections, such that decoder-only LLMs (under reasonable assumptions) can model exactly one sequence in each collection. We leverage this continuity result to provide a proof of why perplexity fails: specifically, we show that the existence of a long enough sequence the model predicts accurately and confidently implies the existence of another sequence with very low perplexity that the model still fails to predict over.

3. Log-perplexity of wrong next-token predictors can arbitrarily approach zero

For the specific case of autoregressive models trained on next-token prediction (such as large language models), we can recombine a few previous results to theoretically strengthen the empirical finding of Fang et al. (2025).

3.1. Preliminaries

As a clean proxy to the points we are going to make, throughout this section we will focus on a task where both perplexity and correctness are easy to define. Specifically, we study the **bitstring copy task**: a language model is provided a sequence of bits followed by a unique "stop" symbol, |, after which it needs to reproduce the given sequence of bits exactly. For example, given `01010|`, the model needs to output `01010`. The model's vocabulary is hence made up of only three symbols: `0`, `1` and |. It is well known that copying is tricky for modern LLMs to learn robustly (Barbero et al., 2024), making it an ideal candidate for our study.

Secondly, all of our results rely on the assumption that our language model, T, is a decoder-only Transformer with compact position embeddings (CPE); exactly matching the assumptions of Pasten et al. (2025), whose key results we rely on. These assumptions are generally true for the majority of positional embeddings in common use today, such as RoPE (Su et al., 2024). We denote the output probability distribution of T as:

$$T(\mathbf{x})(y) = P_T(y \mid \mathbf{x}), \tag{1}$$

the probability of emitting symbol y given input prompt $\mathbf{x}$.

3.2. Deterministic sampling

In order to make robust claims about a model's accuracy and perplexity, we need to assume it will behave **deterministically** across all possible input prompts. We hence assume that its outputs are sampled via *greedy decoding*:

$$T_!(\mathbf{x}) = \arg\max_{s \in \{0,1\}} T(\mathbf{x})(s) \tag{2}$$

assuming all ties are broken consistently, e.g. by always choosing `0` in such cases.

In this regime, we will always measure the **log-perplexity** of the language model, T, on the length-n input bitstring $\mathbf{b} \in [0,1]^n$, defined as follows:

$$\mathrm{pplx}_T(\mathbf{b}) = -\frac{1}{n}\sum_{k=1}^{n} \log T(b_1 \cdots b_n | o_1 \cdots o_{k-1})(b_k), \tag{3}$$

where the symbols o_i are sampled deterministically:

$$o_1 = T_!(b_1 \cdots b_n|) \qquad o_i = T_!(b_1 \cdots b_n | o_1 \cdots o_{i-1}). \tag{4}$$

This aligns well with the model's loss function, and it is monotonically related to the perplexity.

Finally, we assume that the model performs all of its computations with appropriate numerical protection, meaning that the obtained values of $\log T(\mathbf{x})(y)$ will never diverge to $-\infty$, and remain bounded by $\log T(\mathbf{x})(y) \geq \log \varepsilon$ for some $\varepsilon > 0$.

Lemma 3.1 (Perplexity convergence)**.** *Let T be a decoder-only Transformer with compact position embeddings (CPE), as defined by Pasten et al. (2025). Assume T is trained to perform a copy task over bitstrings, and it samples outputs by greedy decoding.*

Let $\boldsymbol{\alpha} = \alpha_1\alpha_2\cdots\alpha_n\cdots$ be an infinite bitstring. Assume T is capable of correctly copying every finite prefix of $\boldsymbol{\alpha}$; that is, there is an $\epsilon > 0$ such that, for all $n \in \mathbb{N}$ and $1 \leq k \leq n$:

$$T(\alpha_1 \cdots \alpha_n | \alpha_1 \cdots \alpha_{k-1})(\alpha_k) > 1/2 + \epsilon. \tag{5}$$

*Then, for every $\xi > 0$, there must exist $n' \in \mathbb{N}$ such that, for all prefixes $\boldsymbol{\alpha}_N = \alpha_1\alpha_2\cdots\alpha_N$ with $N \geq n'$, there is a bitstring $\boldsymbol{\beta}_N$ such that $|\mathrm{pplx}_T(\boldsymbol{\alpha}_N) - \mathrm{pplx}_T(\boldsymbol{\beta}_N)| < \xi$, and $\boldsymbol{\beta}_N$ is **not** correctly copied by T.*

We prove Lemma 3.1 in Appendix A. As it relies on a strong assumption over the "anchor" bitstring $\boldsymbol{\alpha}$, the reader might be curious as to how often this assumption can be met. Interestingly, we can show that only *six* bitstrings are possible choices for $\boldsymbol{\alpha}$ (see Appendix B), but this will combinatorially explode with more symbols in the vocabulary.

Armed with this result, we can now introduce an assumption of T having a certain (high) confidence in copying $\boldsymbol{\alpha}_N$, which will shortly bring us to one of our key results.

Proposition 3.2 (Collapsing confidence)**.** *Let T be a decoder-only Transformer with compact position embeddings (CPE), as defined by Pasten et al. (2025). Assume T is trained to perform a copy task over bitstrings, and it samples outputs by greedy decoding.*

*Let $\boldsymbol{\alpha} = \alpha_1\alpha_2\cdots\alpha_n\cdots$ be an infinite bitstring. Assume T is capable of correctly copying every finite prefix of $\boldsymbol{\alpha}$ with **confidence** $(1-\gamma)$; that is, there is a $0 \leq \gamma < 1/2$ such that, for all $n \in \mathbb{N}$ and $1 \leq k \leq n$:*

$$T(\alpha_1 \cdots \alpha_n | \alpha_1 \cdots \alpha_{k-1})(\alpha_k) \geq 1 - \gamma. \tag{6}$$

*Then, for every $\epsilon > 0$, there must exist $n' \in \mathbb{N}$ such that, for every size $N \geq n'$, there is a bitstring $\boldsymbol{\beta}_N = \beta_1 \cdots \beta_N$ such that $\mathrm{pplx}_T(\boldsymbol{\beta}_N) < -\log(1-\gamma) + \epsilon$, and $\boldsymbol{\beta}_N$ is **not** correctly copied by T.*

Proof. This result can be derived by applying Lemma 3.1, setting $(\epsilon = \frac{1}{2} - \gamma,\ \xi = \epsilon)$, and remarking that $\mathrm{pplx}_T(\boldsymbol{\alpha}_N) \leq -\frac{1}{N}\sum_{k=1}^{N} \log(1-\gamma) = -\log(1-\gamma)$. □

Corollary 3.3. *If there exists any infinite sequence $\boldsymbol{\alpha}$ copied with certainty ($\gamma = 0$) by T, then there must exist a family of finite sequences $\boldsymbol{\beta}_N$, such that $\lim_{N\to+\infty} \mathrm{pplx}_T(\boldsymbol{\beta}_N) = 0$, and none of the sequences in $\boldsymbol{\beta}_N$ are correctly copied by T.*

This result demonstrates that, as models get more confident on any input, this necessarily allows for confounding situations where some other inputs get incorrectly processed without a visible impact on perplexity.

3.3. Stochastic sampling

One important assumption that allowed for this result to be cleanly derived is greedy decoding (i.e. sampling with temperature $\theta = 0$). As this setup is less common in contemporary use of decoder-only Transformers, here we briefly remark on the applicability of our theoretical results in the stochastic sampling case. In our context, increasing θ also increases the likelihood of a "random bit-flip" which would lead to incorrect copying of the bitstring $\boldsymbol{\alpha}_N$.

First, we abstract away the choice of θ by folding it into γ:

Remark 3.4. Let $T(\mathbf{a})(\sigma) = (1-\gamma)$ for an input bitstring $\mathbf{a}$ and bit $\sigma \in \{0,1\}$. Then, assuming we sample with temperature $\theta > 0$, the sampling probability becomes:

$$T_\theta(\mathbf{a})(\sigma) = \frac{(1-\gamma)^{1/\theta}}{(1-\gamma)^{1/\theta} + \gamma^{1/\theta}} = 1 - \gamma', \tag{7}$$

where γ' is a function of γ and θ. Therefore, varying temperature of a $(1-\gamma)$-confident model may be seen as a model with $\theta = 1$ but a different confidence level $(1-\gamma')$. Hence, we may assume $\theta = 1$ without loss of generality.

Firstly, we recall a useful result – Boole's inequality – which allows us to place a meaningful bound on the probability of bit-flips in a $(1-\gamma)$-confident sampler:

Remark 3.5. Assume T is capable of correctly copying a length-N bitstring $\boldsymbol{\alpha}_N$ with confidence $(1-\gamma)$. Then, we can bound the probability of any stochastic copying errors using Boole's inequality (letting $\bar{\alpha}_k = 1 - \alpha_k$):

$$\begin{aligned} 1 - P_T(\boldsymbol{\alpha}_N \mid \boldsymbol{\alpha}_N|) &\leq \sum_{k=1}^{N} T(\alpha_1 \cdots \alpha_N | \alpha_1 \cdots \alpha_{k-1})(\bar{\alpha}_k) \\ &\leq \sum_{k=1}^{N} \gamma = N\gamma. \end{aligned}$$

That is, if $N\gamma \ll 1$, it is unlikely any flips will happen, in which case the baseline sequence $\boldsymbol{\alpha}_N$ is copied correctly.

In the stochastic sampling regime, we can leverage the results of Pasten et al. (2025) once again, to analyse the probability that T will produce $\boldsymbol{\alpha}_N$ in response to $\boldsymbol{\beta}_N$!

Proposition 3.6. *Assume T is capable of correctly copying every finite prefix of $\boldsymbol{\alpha}$ with confidence $(1-\gamma)$. Then, for every $\epsilon > 0$ there is an $n' \in \mathbb{N}$ such that, for every size $N \geq n'$, under stochastic output sampling,*

$$1 - P_T(\boldsymbol{\alpha}_N \mid \boldsymbol{\beta}_N|) \leq N(\gamma + \epsilon), \tag{8}$$

where $\boldsymbol{\beta}_N$ is derived as in Proposition 3.2.

Informal proof. Follow a similar argument to Remark 3.5, but this time consider $N > \lceil 1/\delta \rceil$, where $\delta > 0$ is the continuity condition for ϵ, and leverage continuity. □

This result implies that there are three possible outcomes in the stochastic sampling scenario (where ϵ_N is the smallest value of ϵ attainable at size N):

$N\gamma \ll 1, N\epsilon_N \ll 1$: Corresponds well to our greedy-decoding analysis. The model is confident enough to copy the baseline sequence $\boldsymbol{\alpha}_N$ with high probability, but it's too tethered to $\boldsymbol{\alpha}_N$ (due to continuity). Therefore it will fail to copy $\boldsymbol{\beta}_N$ with high probability (most likely producing $\boldsymbol{\alpha}_N$).

$N\gamma \ll 1, N\epsilon_N \not\ll 1$: The model copies $\boldsymbol{\alpha}_N$ with high probability, and the sequence is not long enough for our theory to apply. In this case, we are unable to make concrete claims about the model's behaviour on $\boldsymbol{\beta}_N$.

$N\gamma \not\ll 1$: The model is not confident enough to reliably copy the baseline sequence $\boldsymbol{\alpha}_N$, and due to continuity, it will likely fail to copy $\boldsymbol{\beta}_N$ in the same way.

3.4. Implications on learning dynamics

In Lemma 3.1, we showed that the perplexity of an incorrect sequence $\boldsymbol{\beta}_N$ converges to that of a correct sequence $\boldsymbol{\alpha}_N$ within a margin ϵ. We now show that this has learnability implications. In particular, as the loss on $\boldsymbol{\alpha}_N$ goes to 0, this implies that the loss on $\boldsymbol{\beta}_N$ also approaches 0. Consequently, the training signal for the incorrect sample vanishes, and such a sample cannot be jointly learned (proved in Appendix C).

Corollary 3.7 (Vanishing gradients on incorrect samples)**.** *Let $\mathcal{L}(\mathbf{x}; T_\theta) = -\frac{1}{M}\sum_{i=1}^{M} \log T_\theta(\mathbf{x}_{<i})(x_i)$ be the standard autoregressive cross-entropy loss for a CPE decoder-only Transformer with parameters θ.*

*Assume that for the sequence $\boldsymbol{\alpha}_N$, the model achieves a perfect loss, i.e. $\mathcal{L}(\boldsymbol{\alpha}_N; T_\theta) \to 0$ as $N \to +\infty$. Under the conditions of Proposition 3.2, for the sequence $\boldsymbol{\beta}_N$ which is **not** correctly copied by T_θ, the gradient of the loss with respect to θ vanishes:*

$$\lim_{N \to +\infty} \|\nabla_\theta \mathcal{L}(\boldsymbol{\beta}_N; T_\theta)\| = 0. \tag{9}$$

3.5. Empirical analysis

We attempt to validate our theoretical results in Figure 2, both when pre-training a CPE Transformer on solely the copy task, and on a larger, general Gemma 3 4B model, setting $\boldsymbol{\alpha}_N = 00\cdots00$ and $\boldsymbol{\beta}_N = 00\cdots01$. Our observations match our expectations: continuity holds in both regimes, with the gap between the probability distributions on $\boldsymbol{\alpha}_N$ and $\boldsymbol{\beta}_N$ diminishing with increasing N. Further, the probability of continuing $\boldsymbol{\alpha}_N$ remains high and stable for most input sizes, whereas the probability of successfully continuing $\boldsymbol{\beta}_N$ collapses. All the while, (log-)perplexity indeed gets iteratively closer between the two sequences.

One important caveat with these results is the observed noisy

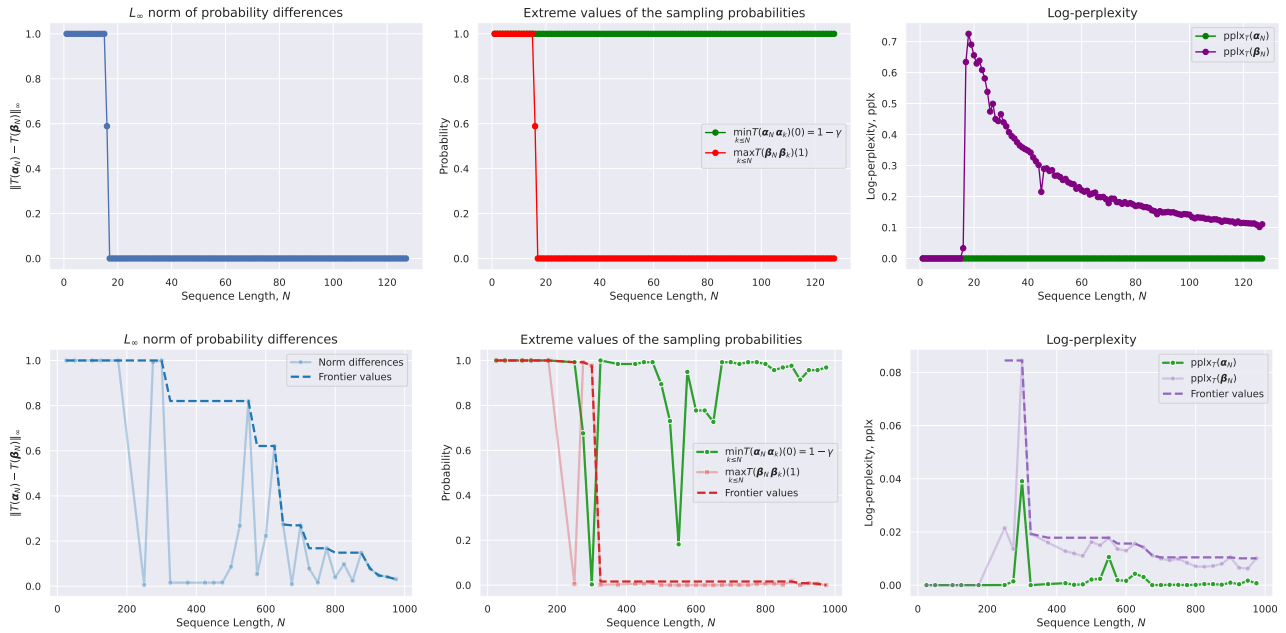


Figure 2. For various sequence lengths, N, on the copy task, we compute (**Left**) the L_∞ norm of the difference between the logit distributions across all positions, (**Middle**) the minimal observed probability of predicting $\boldsymbol{\alpha}_k$—our conservative estimate of $1-\gamma$—and the maximal observed probability of predicting $\boldsymbol{\beta}_N$—which can serve as a bound on the probability that the model will copy $\boldsymbol{\beta}_N$ properly. We also plot (**Right**) the log-perplexity for both $\boldsymbol{\alpha}_N$ and $\boldsymbol{\beta}_N$. This is done both for (**Top**) a toy copy environment where a CPE Transformer is trained on sizes up to 16 bits, and (**Bottom:**) prompting Gemma 3 4B with a copy request.

patterns in the Gemma 3 4B experiments. This is due to the fact that, unlike the clear-cut bitstring vocabulary of our theoretical setup, Gemma 3 has a much larger set of possible tokens—and especially due to their failure to count (Barbero et al., 2024), on certain occasions the model attempts to prematurely predict newline characters and end-of-turn characters. These both cause issues with the computed probability distribution and perplexities, but they do not affect the overall trends of the relevant metrics collapsing, which we visualised using dashed lines.

Almost none of the results derived so far are specific to perplexity's pointwise form—they, instead, mainly rely on the *averaging* process of the equation. As such, one might be tempted to see this as further evidence of Fang et al. (2025)'s claim that the perplexity function itself might not be inherently problematic—just the way it's aggregated. Furthermore, the LongPPL replacement for perplexity which is proposed in Fang et al. (2025) would not necessarily suffer from the smoothing effects we identify here, as it would significantly shorten the number of tokens for which the metric is computed. That said, we believe there *are* inherent issues in the perplexity function beyond how it's averaged, and this motivates us to study a *pointwise* setup with only one output, but placing important emphasis on the model confidence values.

4. An analytic view into confidence

From the theoretical and empirical analysis we presented so far, one variable that clearly stands out is *confidence*. In systems relying on stochastic sampling, high confidence (low γ) is important for them to generalise mechanistically—as $N\gamma$ needs to be sufficiently small to attenuate the likelihood of failures over long ranges N. However, our theory implies that any high-confidence prediction in CPE Transformers not only opens the door to guaranteed failures elsewhere, it does so in a way that perplexity may not be able to detect the failure. In what follows, we attempt to answer: Can we establish a more general connection between the level of confidence a model has and its predictive power, in a way that reveals how predictable that power is via perplexity?

The answer is affirmative—in that, whenever confidence of a model increases, the model needs to supplement that confidence with a sufficient boost in predictive power—otherwise, perplexity will be unable to recognise this jump in confidence as positive. Specifically, with the right set of initial assumptions, it is possible to *analytically* solve for the "critical accuracy" needed to justify increased confidence.

4.1. Preliminaries

In order to be able to analytically manipulate the expressions we care about, it is important to make simplifying

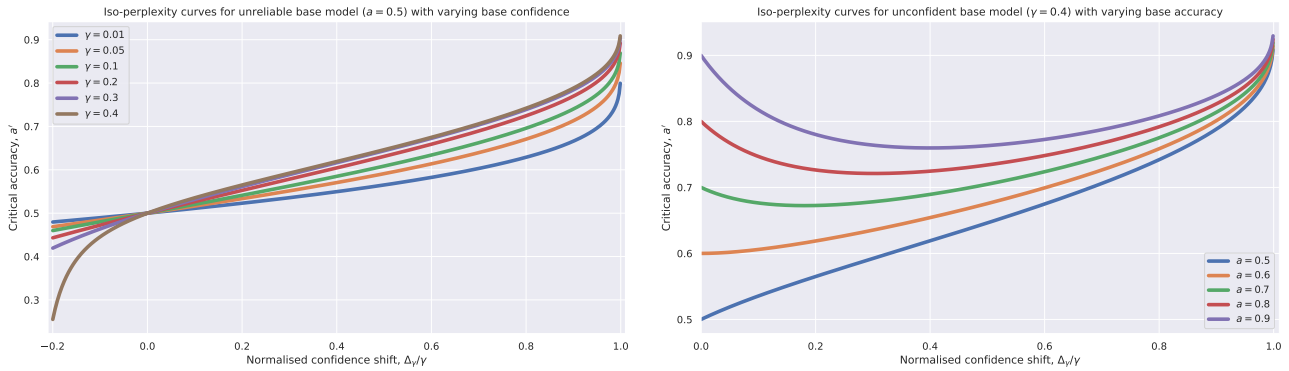


Figure 3. **Left:** Iso-perplexity curves for the setting with an unreliable base model ($a = 0.5$) for varying choices of confidence ($1 - \gamma$). **Right:** Iso-perplexity curves for an unconfident base model ($\gamma = 0.4$) for varying choice of base accuracy a,.

assumptions that will allow us to abstract away our model's *confidence* $(1 - \gamma)$ and *accuracy* (a) as scalar variables in $[0, 1]$. Further, there must be a simple way to relate those scalar variables to the model's log-perplexity, pplx.

The specific framework we assume which captures this idea well is a binary classification problem, where the model *always* makes its decisions with identical confidence $(1-\gamma)$. This means that we can express the log-perplexity over a dataset with accuracy a as:

$$\mathrm{pplx}_{a,\gamma} = -a \log(1 - \gamma) - (1 - a) \log \gamma \qquad (10)$$

4.2. Iso-perplexity curves

Now, consider a setting where the model gets *more confident* by $\Delta_\gamma \in [0, \gamma]$; that is, its confidence when correct jumps to $1 - \gamma + \Delta_\gamma$, and its confidence in the correct answer when wrong drops, symmetrically, to $\gamma - \Delta_\gamma$. If the accuracy doesn't change, the first term of pplx will decrease while the second will increase.

But, even though we symmetrically altered the confidence by Δ_γ, the change in these two terms is *not* symmetrical, as the $\log$ function has a substantially varying rate of change between $(0, 1]$. If we keep accuracy the same when increasing confidence, this will in many cases *increase* perplexity.

Accordingly, a sufficient *rise in accuracy*, a', is needed to compensate for this increase in confidence, if we wish perplexity to recognise this improvement in model accuracy.

This "critical point" in accuracy happens when the perplexities of the old and new model become equal:

$$\begin{aligned} \mathrm{pplx}_{a,\gamma} &= -a \log(1 - \gamma) - (1 - a) \log \gamma \\ &= -a' \log(1 - \gamma + \Delta_\gamma) - (1 - a') \log(\gamma - \Delta_\gamma) \end{aligned}$$

Rearranging the terms, we can derive the required accuracy:

$$a' = \frac{\mathrm{pplx}_{a,\gamma} + \log(\gamma - \Delta_\gamma)}{\log(\gamma - \Delta_\gamma) - \log(1 - \gamma + \Delta_\gamma)} \qquad (11)$$

Note that this function depends on both the initial accuracy a and initial confidence $(1 - \gamma)$. Therefore, it gives rise to several types of *iso-perplexity curves*, depending on whether we keep a fixed and vary γ, or keep γ fixed and vary a. To illustrate what these curves teach us about the reliability of perplexity as a discriminative metric, we focus on two specific cases here:

Iso-perplexity at $a = 0.5$ In this setting, we assume starting from an entirely unreliable model, but varying the starting confidence, $(1 - \gamma)$. We plot the *critical accuracy*, a', against the *normalised confidence shift*, Δ_γ/γ; see Figure 3 (Left). Note that any model falling under the iso-perplexity curve would *not* be selected as improving perplexity, even though its accuracy may be better than the random chance of the first model – similarly, a model may end up above the iso-perplexity curve even though its accuracy is worse than the baseline. We can make two key observations:

Firstly, for all considered starting confidences, not all better more-confident models will decrease perplexity. The afforded "breathing room" for a' tends to be greater (in relative terms) the more confident the base model is – there is "less surprise" when making an already confident model more confident. Still, many confidence shifts require increasing accuracy by over 5–10%, which is very significant.

Secondly, no matter what the starting confidence, truly extraordinary confidence requires truly extraordinary evidence—as $\Delta_\gamma \to \gamma$, $a' \to 1$. Put differently, a perfectly confident model must be perfectly accurate, otherwise it will always be rejected by perplexity.

Iso-perplexity at $\gamma = 0.4$ In this setting, we start from an unconfident model, but varying the starting accuracy, a—once again plotting critical accuracy against normalised confidence shift. The resulting iso-perplexity curves are in Figure 3 (Right). The key insight that this regime offers is the existence of "unjustified free lunch" zones, where

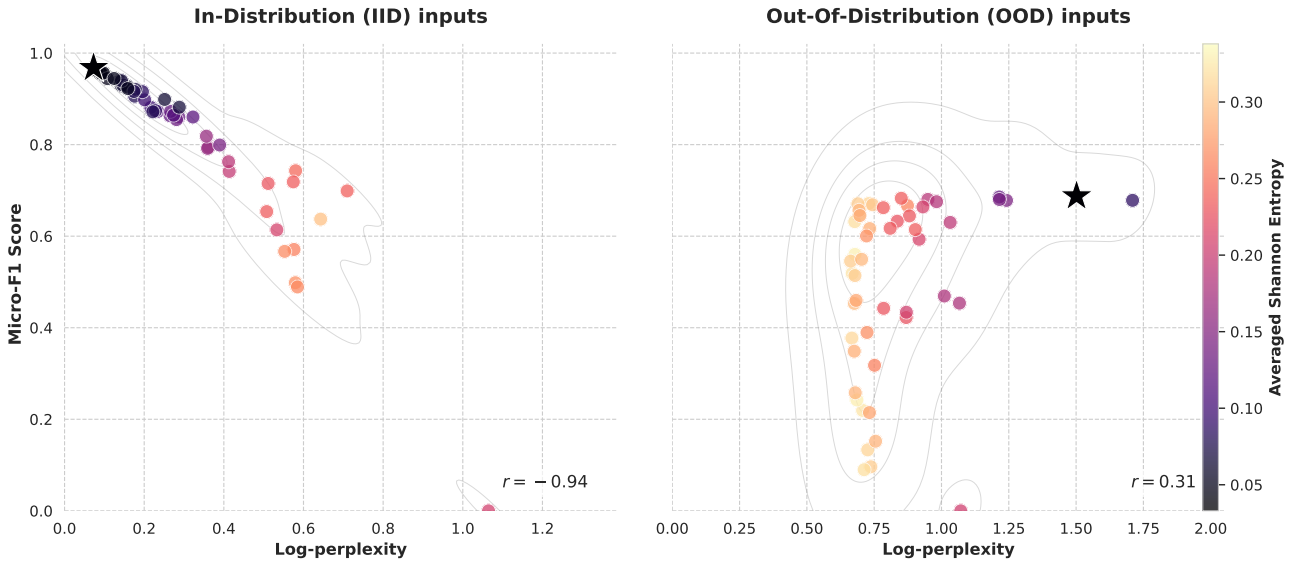


Figure 4. Scatter plots of micro-F_1 scores against log-perplexities, L, for various checkpoints of a Transformer model trained on the Parity problem, as specified by Vitvitskyi et al. (2025), for both in-distribution (**Left**) and out-of-distribution (**Right**) held-out data. We also colour-code the checkpoints by their averaged Shannon entropy, $\bar{H}$, provide the Pearson correlation coefficient, r, and highlight the point with the highest accuracy by using a star (also colour-coded by entropy).

the iso-perplexity curves are decreasing for positive Δ_γ. If an unconfident model is already sufficiently accurate, it is possible to improve their perplexity just by making them more confident – even if this leads to significant drops in accuracy ($a' < a$).

It is evident that there exists a rather non-negligible space under the iso-perplexity curve with $a' > a$, as well as above it with $a' < a$; it describes a significant family of models that would not be selected by perplexity, in spite of being better predictors than their baseline. We hypothesise this might have implications in many relevant regimes of AI deployment where accuracy cannot be easily measured, *especially* as the model needs to predict outside of its training distribution, which often requires higher confidence.

4.3. How often are we on the wrong iso-perplexity side?

Having exposed that there exist very clear regions of the model confidence/accuracy space where perplexity would not select the more accurate model, what remains to be seen is to what extent will this occur in practice.

One might hypothesise that a model is particularly vulnerable to such failures when the inputs stray *out-of-distribution* (OOD) compared to data the model was exposed to during training. Indeed, as we saw with the copy task example, models might *need* to maintain a low value of γ in order to even be able to process their baseline sequences properly. However, higher confidence also implies that when any failures do occur, they will be especially painful to perplexity.

Note that our "identical-confidence" model described in Equation 10 is substantially *constrained* in order to make iso-perplexity curves analytically derivable—in reality, there may well not be a value of γ that fits an observed perplexity/accuracy pair (L, a) over a real dataset. That is, there often may be no $\gamma \in [0, 1]$ such that $L = -a \log(1-\gamma) - (1-a)\log\gamma$. Furthermore, if any models start to get less confident, iso-perplexity curves approach their singularity point at $\gamma = 0.5$, at which point it gets hard to see the relevant phase transitions on the confidence/accuracy plot.

For all of the above reasons, we abandon plotting the iso-perplexities here, and instead directly plot the (L, a) pairs we observed via a scatter plot. Whenever $L_1 < L_2$ but $a_1 < a_2$, the model's perplexity cannot discriminate properly between these two points, and we can estimate how often this happens by observing the Pearson correlation coefficient, r. In an ideal setting, where the L metric exactly orders accuracies, we would recover $r \approx -1$.

Beyond measuring the frequency of incorrect model selections, we also want to ascertain that these issues can be directly related to model confidence. While we cannot map arbitrary logit collections $\{\log p_i\}_{i=1}^n$ to a fixed value of γ, we *can* compute a proxy for the model's overall level of certainty by computing the *averaged Shannon entropy*, $\bar{H} = -\frac{1}{n}\sum_{i=1}^n p_i \log p_i$. We can then use this quantity to colour-code the individual models we're studying on the scatter plot—under the assumption that points that will be particular outliers in the OOD setting are the ones where $\bar{H}$ is lower (when the model is on the whole more confident).

4.4. Parity task setup

When deciding on which problem to choose to study these effects, it is not only desirable for the task to have a natural OOD regime—it should also be seen as "mechanistically easy but practically hard". By this, we mean that there is a very clear, simple procedure that generates the ground-truth outputs, yet it is known that reproducing those outputs is hard for contemporary AI systems. The existence of a clear target procedure means that models need to get confident in order to replicate this procedure; the practical hardness means that their confidence will not always be rewarded.

A very good fit is the **parity** task: given a bitstring, predict the exclusive-or (XOR) of all of its bits (e.g., for `01010`, predict `0`; for `11010`, predict `1`). Parity is well-understood to be difficult when length-generalising with Transformers, for known theoretical reasons (Hahn, 2020), yet the target formula for computing the output is very simple.

We replicate the Transformer training setup for the Parity task from Vitvitskyi et al. (2025), reusing the baseline hyperparameters leveraged there, and training the model for $5,000$ gradient steps. Our aim is to appreciate how the model's performance/confidence profile evolves throughout training, and hence, we save many checkpoints of the model throughout training—one for every 100 steps of gradient descent taken—and evaluate them on held-out in-distribution (IID) and out-of-distribution (OOD) bitstrings in terms of size. In this case, we train on bitstrings of size up to 16, and consider an OOD distribution of bitstrings of size 128.

Overall, this procedure generates a dataset of $(L, \mathrm{F}_1, \bar{H})$ tuples, where L is (log-)perplexity, F_1 is the micro-F_1 score obtained by the model on those sequences, and $\bar{H}$ is the averaged Shannon entropy estimated using the model's individual parity prediction logits across the entire bitstring.

4.5. Results and Discussion

We visualise the corresponding scatter plots of the stored checkpoints, along with other useful data (colour-coding, Pearson correlation) in Figure 4. We find that these visualisations provide strong evidence for our hypothesis, by making the following observations over the two data distributions:

Training progression While in-distribution the model appears to gradually improve its loss and performance, with a corresponding decrease in entropy as the model gets more confident, the same trajectory cannot be observed in the OOD case. Worse yet, the checkpoint with the optimal OOD accuracy is one of the worst in terms of OOD perplexity.

Pearson correlation The IID evaluations of the checkpoints paint a picture of a model whose perplexity improvements, for the most part, track micro-F_1 score improvements; indeed, with $r = -0.94$, there is a strong anticorrelation between the two variables. No such trend can be observed OOD, in fact, the empirical value of r is *positive* rather than negative. This neatly translates to the likelihood that we have a pair of incorrectly discriminated points with $L_1 < L_2$ but $a_1 < a_2$: it is very high in the OOD regime.

Entropy connection Lastly, the entropy colour-coding reveals the final piece of the puzzle and matches our hypothesis very well. In-distribution, entropy reduction is a sign of model maturity: the confidence increase follows a clear jump in predictive power and decrease in loss. Out-of-distribution, however, the checkpoints with low entropy can retain predictive power while drastically harming perplexity. In fact, the aforementioned best-performing observed OOD model has one of the lowest entropies in the entire dataset.

All taken together, we can make a clear conclusion: in the right kind of *out-of-distribution* regime, *many* points end up on the *wrong* side of iso-perplexity, and this effect can be directly tied to an *increase in confidence*.

5. Conclusions

In their recent important work, Fang et al. (2025) make a clear stance on the issues behind perplexity on long ranges:

"... there is growing evidence that LLMs' perplexity does not indicate their performance on long-context benchmarks. There are two possible sources of this mismatch: either the log-likelihood-based metric is flawed, or the averaged tokens are not representative enough. In this work, we champion the latter explanation..."

We provided theoretical evidence in support of the latter source—with all tokens contributing to an averaged loss, this has the potential to lead to *weird* situations, where a model makes confident mistakes on an input, yet its log-perplexity can get arbitrarily close to zero for that input.

However, we also found that the former source cannot be ignored—the perplexity metric itself is inherently skewed, and prone to favouring less confident predictors, *especially* in the long-context settings mentioned above. We found that high model confidence, coupled with a perplexity objective, can be the very reason for being able to construct the weird situations in the former paragraph. We provided additional evidence for this by studying the ample unfavourable regions with respect to *iso-perplexity curves*.

While we do not offer an alternative to perplexity in regimes where accuracy cannot be measured, we hope that our work serves as a useful foundation for exercising appropriate care when using perplexity, as well as offering a few "diagnostic approaches" that can help us estimate in which situations one might need to rethink their model selection protocol.

References

Barbero, F., Banino, A., Kapturowski, S., Kumaran, D., Madeira Araújo, J., Vitvitskyi, O., Pascanu, R., and Veličković, P. Transformers need glasses! information over-squashing in language tasks. *Advances in Neural Information Processing Systems*, 37:98111–98142, 2024.

Fang, L., Wang, Y., Liu, Z., Zhang, C., Jegelka, S., Gao, J., Ding, B., and Wang, Y. What is wrong with perplexity for long-context language modeling? In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=fL4qWkSmtM.

Gelberg, Y., Eguchi, K., Akiba, T., and Cetin, E. Extending the context of pretrained llms by dropping their positional embeddings. *arXiv preprint arXiv:2512.12167*, 2025.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *International conference on machine learning*, pp. 1321–1330. PMLR, 2017.

Hahn, M. Theoretical limitations of self-attention in neural sequence models. *Transactions of the Association for Computational Linguistics*, 8:156–171, 2020.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Hsieh, C.-P., Sun, S., Kriman, S., Acharya, S., Rekesh, D., Jia, F., Zhang, Y., and Ginsburg, B. Ruler: What's the real context size of your long-context language models? *arXiv preprint arXiv:2404.06654*, 2024.

Hu, Y., Huang, Q., Tao, M., Zhang, C., and Feng, Y. Can perplexity reflect large language model's ability in long text understanding? In *The Second Tiny Papers Track at ICLR 2024*, 2024. URL https://openreview.net/forum?id=Cjp6YKVeAa.

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1xwNhCcYm.

Pasten, H., Urrutia, F., Orellana, H. I. J., Calderon, C. B., Rojas, C., and Kozachinskiy, A. Continuity and isolation lead to doubts or dilemmas in large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=dR58v9Dd42.

Peng, B., Quesnelle, J., Fan, H., and Shippole, E. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

Rabin, M. O. and Scott, D. Finite automata and their decision problems. *IBM journal of research and development*, 3(2):114–125, 1959.

Ranzato, M., Chopra, S., Auli, M., and Zaremba, W. Sequence level training with recurrent neural networks. In *ICLR (Poster)*, 2016. URL http://arxiv.org/abs/1511.06732.

Schmidt, F. Generalization in generation: A closer look at exposure bias. In Birch, A., Finch, A., Hayashi, H., Konstas, I., Luong, T., Neubig, G., Oda, Y., and Sudoh, K. (eds.), *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pp. 157–167, Hong Kong, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5616. URL https://aclanthology.org/D19-5616/.

Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Theis, L., Oord, A. v. d., and Bethge, M. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Vitvitskyi, A., Araújo, J. G., Lackenby, M., and Veličković, P. What makes a good feedforward computational graph? *arXiv preprint arXiv:2502.06751*, 2025.

Wang, C. and Sennrich, R. On exposure bias, hallucination and domain shift in neural machine translation. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3544–3552, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.326. URL https://aclanthology.org/2020.acl-main.326/.

A. Proof of Lemma 3.1.

Lemma A.1 (3.1.: Perplexity convergence)**.** *Let T be a decoder-only Transformer with compact position embeddings (CPE), as defined by Pasten et al. (2025). Assume T is trained to perform a copy task over bitstrings, and it samples outputs by greedy decoding.*

Let $\boldsymbol{\alpha} = \alpha_1\alpha_2\cdots\alpha_n\cdots$ be an infinite bitstring. Assume T is capable of correctly copying every finite prefix of $\boldsymbol{\alpha}$; that is, there is an $\epsilon > 0$ such that, for all $n \in \mathbb{N}$ and $1 \leq k \leq n$:

$$T(\alpha_1\cdots\alpha_n|\alpha_1\cdots\alpha_{k-1})(\alpha_k) > 1/2 + \epsilon. \tag{12}$$

*Then, for every $\xi > 0$, there must exist $n' \in \mathbb{N}$ such that, for all prefixes $\boldsymbol{\alpha}_N = \alpha_1\alpha_2\cdots\alpha_N$ with $N \geq n'$, there is a bitstring $\boldsymbol{\beta}_N$ such that $|\mathrm{pplx}_T(\boldsymbol{\alpha}_N) - \mathrm{pplx}_T(\boldsymbol{\beta}_N)| < \xi$, and $\boldsymbol{\beta}_N$ is **not** correctly copied by T.*

Proof. Since T is a CPE decoder-only Transformer, by Pasten et al. (2025)'s continuity theorem, there must exist $\delta > 0$ such that, for any two equal-length inputs $\mathbf{x}$ and $\mathbf{x}'$, if their relativised Hamming distance $d_H(\mathbf{x}, \mathbf{x}') < \delta$ and their last symbol is identical, then $\|T(\mathbf{x}) - T(\mathbf{x}')\|_\infty \leq \epsilon$.

Whenever $n_c > \lceil 1/\delta \rceil$, we can find a $\boldsymbol{\beta}_{n_c}$ such that $d_H(\boldsymbol{\alpha}_{n_c}, \boldsymbol{\beta}_{n_c}) < \delta$—simply flip exactly one bit in $\boldsymbol{\alpha}_{n_c}$ at an arbitrary position, j. Coupled with Equation 5's assumption, we can deduce

$$T(\beta_1\cdots\beta_{n_c}|\alpha_1\cdots\alpha_{k-1})(\alpha_k) > 1/2, \tag{13}$$

therefore,

$$T_!(\beta_1\cdots\beta_{n_c}|) = \alpha_1 \qquad T_!(\beta_1\cdots\beta_{n_c}|\alpha_1\cdots\alpha_{k-1}) = \alpha_k \tag{14}$$

for all $1 \leq k \leq n_c$. That is, $\boldsymbol{\beta}_{n_c}$ is **not** correctly copied by T, and its copying log-perplexity is:

$$\mathrm{pplx}_T(\boldsymbol{\beta}_{n_c}) = -\frac{1}{n_c}(\log T(\beta_1\cdots\beta_{n_c}|\alpha_1\cdots\alpha_{j-1})(\beta_j) + \sum_{k\neq j}\log T(\beta_1\cdots\beta_{n_c}|\alpha_1\cdots\alpha_{k-1})(\alpha_k)). \tag{15}$$

Now, once we observe that we can also, analogously, express

$$\mathrm{pplx}_T(\boldsymbol{\alpha}_{n_c}) = -\frac{1}{n_c}(\log T(\alpha_1\cdots\alpha_{n_c}|\alpha_1\cdots\alpha_{j-1})(\alpha_j) + \sum_{k\neq j}\log T(\alpha_1\cdots\alpha_{n_c}|\alpha_1\cdots\alpha_{k-1})(\alpha_k)), \tag{16}$$

we can match the relevant terms in the summations to obtain $|\mathrm{pplx}_T(\boldsymbol{\alpha}_{n_c}) - \mathrm{pplx}_T(\boldsymbol{\beta}_{n_c})| \leq -\frac{1}{n_c}(\log(\frac{1}{2}+\epsilon) - \log\varepsilon + (n_c-1)\epsilon)$. By algebraic manipulation of this expression we can conclude that, as long as we choose $n_c > \frac{\epsilon - \log(\frac{1}{2}+\epsilon) + \log\varepsilon}{\xi+\epsilon}$, it will hold that $|\mathrm{pplx}_T(\boldsymbol{\alpha}_{n_c}) - \mathrm{pplx}_T(\boldsymbol{\beta}_{n_c})| < \xi$. This implies that we can set

$$n' = \max\left(\underbrace{\lceil 1/\delta \rceil}_{\text{continuity}}, \underbrace{\frac{\epsilon - \log\left(\frac{1}{2}+\epsilon\right) + \log\varepsilon}{\xi+\epsilon}}_{\text{oversmoothing}}\right), \tag{17}$$

at which point we are guaranteed to obtain both the effects of continuity, misclassifying $\boldsymbol{\beta}_{n'}$, *and* smoothing out the perplexity spike obtained by that misclassification. □

B. How many anchor (bit)strings for Lemma 3.1 exist?

Lemma 3.1 relies on a very strong requirement that the "anchor bitstring" $\boldsymbol{\alpha}$ is predicted correctly for every possible prefix $\boldsymbol{\alpha}_N$. In this Appendix we analyse how many possible choices of $\boldsymbol{\alpha}$ exist, finding potentially surprising results.

Lemma B.1. *There exist only **six** infinite bitstrings $\boldsymbol{\alpha}$ for which there exists a CPE decoder-only Transformer T, capable of correctly copying every finite prefix $\boldsymbol{\alpha}_N$; i.e., there existing $\epsilon > 0$ such that for all n and $1 \leq k \leq n$:*

$$T(\alpha_1\cdots\alpha_n|\alpha_1\cdots\alpha_{k-1})(\alpha_k) > 1/2 + \epsilon. \tag{18}$$

Specifically, the six bitstrings are the ones where, for all $k \in \mathbb{N}$, the value of α_k uniquely determines α_{k+1}. These are:

$$\begin{aligned}
\boldsymbol{\alpha}^{(0;\ 0\to0)} &= \texttt{00000000}\cdots \\
\boldsymbol{\alpha}^{(1;\ 1\to1)} &= \texttt{11111111}\cdots \\
\boldsymbol{\alpha}^{(0;\ 0\to1,\ 1\to0)} &= \texttt{01010101}\cdots \\
\boldsymbol{\alpha}^{(1;\ 0\to1,\ 1\to0)} &= \texttt{10101010}\cdots \\
\boldsymbol{\alpha}^{(0;\ 0\to1,\ 1\to1)} &= \texttt{01111111}\cdots \\
\boldsymbol{\alpha}^{(1;\ 0\to0,\ 1\to0)} &= \texttt{10000000}\cdots
\end{aligned}$$

where the notation $\boldsymbol{\alpha}^{(\alpha_1;\ f)}$ denotes the initial bit α_1 and the function $f : \{\texttt{0},\texttt{1}\} \to \{\texttt{0},\texttt{1}\}$ satisfying $f(\alpha_k) = \alpha_{k+1}$.

Proof. Clearly, there exists a CPE decoder-only Transformer that can correctly copy each of the bit-strings $\boldsymbol{\alpha}^{(\alpha_1,f)}$; we can, for example, force all attention masks to use diagonal attention, and computing the corresponding f in the feedforward layers. Since the "effective context" never grows, the same feedforward layers can be reused at arbitrary prefix lengths without any differences in prediction logits coming out. Now, we need to show that no other infinite bitstring, $\boldsymbol{\alpha}'$, can be correctly copied by *any* CPE decoder-only Transformer, T.

To see why, let us assume that $\boldsymbol{\alpha}'$ is not expressible in the "one-step" form above. This means there must exist two positions, k and l, where the determinism breaks. That is, for two bits $b, b' \in \{\texttt{0},\texttt{1}\}$, $\alpha'_k\alpha'_{k+1} = bb'$ and $\alpha'_l\alpha'_{l+1} = b\bar{b'}$. For a given integer $N > |k-l|$, we can then consider asking our transformer, T, to copy prefixes $\boldsymbol{\alpha}'_{N+l}$ and $\boldsymbol{\alpha}'_{N+k}$.

Consider the following partial outputs resulting from this task:

$$\begin{aligned}
\mathbf{p}_1^{(N)} &= \alpha'_1\cdots\alpha'_{N+l}|\alpha'_1\cdots\alpha'_k \\
\mathbf{p}_2^{(N)} &= \alpha'_1\cdots\alpha'_{N+k}|\alpha'_1\cdots\alpha'_l
\end{aligned}$$

For $\boldsymbol{\alpha}'$ to be correctly copied by T, both of these then must hold:

$$T(\alpha'_1\cdots\alpha'_{N+l}|\alpha'_1\cdots\alpha'_k)(\alpha'_{k+1}) > 1/2+\epsilon \qquad T(\alpha'_1\cdots\alpha'_{N+k}|\alpha'_1\cdots\alpha'_l)(\alpha'_{l+1}) > 1/2+\epsilon \tag{19}$$

The length of both of these partial strings is $N+k+l+1$ (including the separator symbol, $|$), and they also end with the same bit (since we assumed $\alpha'_k = \alpha'_l$). Furthermore, consider the relative Hamming distance between the two partial strings, $d_H(\mathbf{p}_1^{(N)}, \mathbf{p}_2^{(N)})$. Since their first N bits must coincide, we can bound $d_H(\mathbf{p}_1^{(N)}, \mathbf{p}_2^{(N)}) \leq \frac{k+l+1}{N+k+l+1}$. Hence, for N sufficiently large, $d_H(\mathbf{p}_1^{(N)}, \mathbf{p}_2^{(N)}) < \delta$ for any given $\delta > 0$.

Taken together, these conditions allow us to invoke Pasten et al. (2025)'s continuity theorem on $\mathbf{p}_1^{(N)}$ and $\mathbf{p}_2^{(N)}$, for any particular gap ϵ. Put differently, for any $\epsilon > 0$, there always exists a critical size $n' \in N$ such that, for all $N \geq n'$,

$$\|T(\mathbf{p}_1^{(N)}) - T(\mathbf{p}_2^{(N)})\|_\infty \leq \epsilon \tag{20}$$

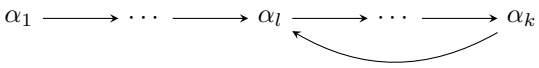
Now, assume that the first of our conditions in Equation 19 holds; that is, $T(\mathbf{p}_1^{(N)})(\alpha'_{k+1}) = T(\mathbf{p}_1^{(N)})(b') > 1/2+\epsilon$. Once N is sufficiently large, due to continuity, $T(\mathbf{p}_2^{(N)})(b') > 1/2$. However, this implies that $T(\mathbf{p}_2^{(N)})(\bar{b'}) = T(\mathbf{p}_2^{(N)})(\alpha'_{l+1}) < 1/2$, violating the second condition! We derived a contradiction, implying that our assumption – that $\boldsymbol{\alpha}'$ cannot be correctly copied by any given CPE decoder-only Transformer, T. □

A very elegant aspect of the above proof is that it naturally generalises to *arbitrary* vocabularies and *finite state machines* defined over them. Specifically:

Remark B.2. Given a vocabulary of size V, the number of infinite strings over it that can be correctly copied by some CPE decoder-only Transformer is given by:

$$S_V = \sum_{k=1}^{V} k \cdot \frac{V!}{(V-k)!} \tag{21}$$

To see why, note that, *eventually*, any string generated with a one-symbol deterministic rule must start cycling (repeating substrings) – essentially at the moment when the first repeated occurs. Further, the substring that then indefinitely cycles must be a *suffix* of the string generated so far. This can be easily visualised by the following generative diagram:



This provides the direct rationale for the formula provided in Equation 21. To enumerate all possible unique computation graphs of this form, we first have to choose the number of states visited, k – noting it cannot be larger than V as all symbols in this prefix must be unique – this induces the sum over all possible $1 \leq k \leq V$. Once we've picked k, we need to choose a permutation of k items; it is known that there are $P(V,k) = \frac{V!}{(V-k)!}$ such permutations. Finally, we have to choose at which position the circular edge "intercepts" the graph and begins cycling, and there are k possible indices where this might happen. This yields the $k \cdot \frac{V!}{(V-k)!}$ possibilities for a particular choice of k, and completes the argument.

As such, even though there are only six *bit*strings that can be correctly copied by CPE decoder-only Transformers, the number of possible strings grows very quickly with the vocabulary size, V. For example, already at $V = 10$, there are $88,776,910$ such strings. We can also observe the process generating these strings as a *deterministic finite automaton* (DFA); from this perspective, the argument in this Remark is very similar to the *pumping lemma* for regular languages (Rabin & Scott, 1959), in the particular case where the observed symbol is made equivalent to the current state.

C. Proof of Corollary 3.7

Corollary C.1 (3.7.: Vanishing gradients on incorrect samples)**.** *Let $\mathcal{L}(\mathbf{x}; T_\theta) = -\frac{1}{M}\sum_{i=1}^{M} \log T_\theta(\mathbf{x}_{<i})(x_i)$ be the standard autoregressive cross-entropy loss for a CPE decoder-only Transformer with parameters θ.*

*Assume that for the sequence $\boldsymbol{\alpha}_N$, the model achieves a perfect loss, i.e. $\mathcal{L}(\boldsymbol{\alpha}_N; T_\theta) \to 0$ as $N \to +\infty$. Under the conditions of Proposition 3.2, for the sequence $\boldsymbol{\beta}_N$ which is **not** correctly copied by T_θ, the gradient of the loss with respect to θ vanishes:*

$$\lim_{N \to +\infty} \|\nabla_\theta \mathcal{L}(\boldsymbol{\beta}_N; T_\theta)\| = 0. \tag{22}$$

Proof. The gradient of the cross-entropy loss with respect to parameters θ is given by

$$\nabla_\theta \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{p}_i - \mathbf{y}_i)^\top J_\theta(\mathbf{x}{<}\, i), \tag{23}$$

where $\mathbf{p}_i = T_\theta(\mathbf{x}_{<i})$ is the predicted probability distribution, $\mathbf{y}_i$ is the one-hot target vector for $\boldsymbol{\beta}_i$, and $J_\theta(\mathbf{x}_{<i}) = \nabla_\theta T_\theta(\mathbf{x}_{<i})$ is the Jacobian of the model logits with respect to the parameters. We can bound the norm of the gradient using the Cauchy-Schwarz inequality:

$$\|\nabla_\theta \mathcal{L}\| \leq \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i - \mathbf{y}_i\| \|J_\theta(\mathbf{x}{<}\, i)\|. \tag{24}$$

As we assume that the Transformer is compact, this implies that the norm of the Jacobian is upper bounded by some K (Lipschitz property), i.e. $\sup_{\mathbf{x}} \|J_\theta(\mathbf{x})\| \leq K$. Furthermore, since the cross-entropy loss is minimised only when the predictive distribution matches the target, convergence in loss implies convergence in the predicted targets:

$$\lim_{N \to +\infty} \|\mathbf{p}_i - \mathbf{y}_i\| = 0 \quad \forall i. \tag{25}$$

Substituting the bounds back, we achieve the desired result:

$$\lim_{N \to +\infty} \|\nabla_\theta \mathcal{L}\| \leq \lim_{N \to +\infty} K \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_i - \mathbf{y}_i\| = 0. \tag{26}$$

□