

Large language models for pattern recognition in text data

Aknur Kossayakova¹, Kurmashev Ildar¹, Luigi La Spada², Nida Zeeshan², Makhabbat Bakyt³,
Moldamurat Khuralay⁴, Omirzak Abdirashev⁴

¹Department of Information and Communication Technologies, M. Kozybayev North Kazakhstan University, Petropavlovsk, Kazakhstan

²School of Computing, Engineering and the Built Environment, Edinburgh Napier University, Edinburgh, United Kingdom

³Department of Information Security, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

⁴Department of Space Technique and Technology, L.N. Gumilyov Eurasian National University, Astana, Kazakhstan

Article Info

Article history:

Received Oct 10, 2024

Revised Oct 18, 2025

Accepted Nov 8, 2025

Keywords:

BERT-base

Computational efficiency

Expected calibration error

GPT-2

In context learning

Large language models

Question answering

ABSTRACT

Large language models (LLMs) are widely deployed in settings where both reliability and efficiency matter. We present a calibrated, seed-robust empirical comparison of an encoder fine-tuned model (bidirectional encoder representations from transformers (BERT)-base) and a decoder in-context model (generative pre-trained transformer (GPT)-2 small) across Stanford question answering dataset v2.0 (SQuAD v2.0) and general language understanding evaluation (GLUE)-multi-genre natural language inference (MNLI), Stanford sentiment treebank 2 (SST-2). Beyond accuracy, we assess reliability (expected calibration error with reliability diagrams and confidence-coverage analysis) and efficiency (latency, memory, throughput) under matched conditions and three fixed seeds. BERT-base yields higher accuracy and lower calibration error, while GPT-2 narrows gaps under few-shot prompting but remains more sensitive to prompt design and context length. Efficiency benchmarks show that decoder-only prompting incurs near-linear latency/memory growth with k-shot exemplars, whereas fine-tuned encoders maintain stable per-example cost. These findings offer practical guidance on when to prefer fine-tuning versus prompting and demonstrate that reliability must be evaluated alongside accuracy for risk-aware deployment.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Makhabbat Bakyt

Department of Information Security, L.N. Gumilyov Eurasian National University

Satpayev str. 2, Astana, Kazakhstan

Email: bakyt.makhabbat@gmail.com

1. INTRODUCTION

Pattern recognition in text data is a fundamental task within natural language processing (NLP) and artificial intelligence (AI). The transformer architecture, a type of hierarchical neural network, has been widely adopted for its effectiveness in modeling long-term dependencies. Generative pre-trained transformer (GPT), an advanced model from OpenAI, exemplifies the potential of transformers in diverse NLP tasks. The introduction of the transformer marked a paradigm shift in NLP, establishing principles like self-attention and bidirectional encoding. While existing literature reviews open-source models like bidirectional encoder representations from transformers (BERT) and GPT-2 (reviewed in section 2), a significant research gap remains: a critical examination of hierarchical pattern recognition mechanisms in opaque, commercially deployed models like GPT. This study explores how large language models (LLMs) recognize patterns, using GPT as a representative case, moving from basic linguistic features to deeper semantic representations [1]–[5]. A primary objective is to analyze the GPT architecture in detail,

examining how design choices, such as stacked transformer layers and multi-head self-attention, affect the model's ability to learn and identify patterns, thereby determining if strong performance stems from architectural innovation as much as from the scale of its training data. This includes an examination of the attention mechanism, which allows simultaneous consideration of multiple linguistic aspects and the formation of hierarchical representations. Another objective is to assess GPT's generalization by evaluating not only reported performance metrics but also the conditions under which generalization succeeds or fails, considering the challenge that reliance on massive data scale poses for reproducibility and computational cost. Finally, the study examines the real-world applications of GPT in domains like healthcare and law, acknowledging that its utility is counterbalanced by its susceptibility to factual inaccuracies, misinterpretations, and inappropriate outputs, which highlights the critical need for responsible deployment that prioritizes accuracy and interpretability.

This work makes the following contributions: i) Calibrated evaluation framework for pattern recognition in text. We introduce a unified framework that evaluates accuracy, reliability (expected calibration error (ECE) with reliability diagrams, confidence–coverage analysis), and efficiency (latency, memory, and throughput) under matched conditions. This framework explicitly contrasts encoder fine tuning with decoder in context learning so that performance is interpreted alongside computational cost; ii) Controlled, seed robust comparison of encoder vs. decoder paradigms. Using public benchmarks (Stanford question answering dataset v2.0 (SQuAD v2.0); general language understanding evaluation (GLUE)—multi-genre natural language inference (MNLI), Stanford sentiment treebank 2 (SST-2)), we run fixed seed experiments (three seeds) to quantify not only accuracy but also calibration and selective prediction behavior. We expose prompt sensitivity effects for the decoder only model and show how these effects interact with calibration; iii) Error taxonomy linked to calibration and abstention. We provide qualitative and quantitative analyses of failure modes (hallucinations/abstention errors, span boundary errors, label confusions) and show that calibration aware thresholds can trade coverage for reliability in deployment scenarios; and iv) Compute aware guidance for practitioners. We delineate regimes where fine-tuned encoders provide stable, compute efficient performance versus regimes where in context decoders are competitive, offering actionable guidance for budget constrained applications.

2. STATE OF THE ART

Transformer-based language models achieve remarkable success by learning hierarchical representations of text. Unlike older sequential architectures, transformers utilize multiple self-attention layers that integrate low-level linguistic features into higher-level abstractions. Empirical analysis shows these models capture tree-like syntactic structures within their latent spaces, aligning with linguistic theories of syntax and semantics. The multi-head self-attention mechanism is central to this hierarchical organization. Multiple heads allow simultaneous analysis of sentence parts, with some heads specializing in syntactic or semantic functions [6]–[11]. Although attention weights offer interpretive clues, they do not fully reveal model reasoning. Nevertheless, a consensus holds that transformers learn rich, structured representations where earlier layers capture lexical features and deeper one's abstract semantics.

Having established hierarchical representations, we examine scaling. Over the past five years, models have expanded dramatically, exemplified by OpenAI's GPT-3 (2020) (175 billion parameters), which demonstrated exceptional few-shot learning performance. This shift from BERT's bidirectional pretraining to GPT-3's large-scale autoregressive framework marked transition from supervised fine-tuning toward prompt-based adaptation. While BERT-style encoders accumulate features for task-specific fine-tuning, large decoder-only models leverage scale for strong zero-/few-shot results. This reorients learning to the prompt, increasing adaptability but also sensitivity to prompt design. Scaling, however, involves trade-offs between data, parameters, and compute. Studies on compute-optimal training show that moderately sized models trained on substantially more tokens can match or exceed larger ones, redefining scaling as balance rather than race to maximal parameter counts [12]. Frameworks such as text-to-text transfer transformer (T5), pathways language model (PaLM), and large language model meta AI (LLaMA) broaden multilingual and multi-task coverage while maintaining a unified interface, enabling more efficient transfer and comparison [13], [14]. To consolidate these developments, Table 1 compares representative transformer families—BERT, GPT-2/3/3.5/4, T5, LLaMA, PaLM, Chinchilla, and Claude—in terms of size, data, benchmarks, and reported limitations, illustrating scaling trends and reliability challenges. Benchmarks listed are representative; exact scores vary by variant and setup. Proprietary models disclose limited training details; values reflect public reports at time of writing.

LLM capabilities are defined by representation hierarchy, scaling efficiency, and reliability/interpretability. Models like T5 (11 B), LLaMA (2023, up to 65 B), Claude, PaLM, and Chinchilla optimize the size-data trade-off. Larger, more diverse training data generally correlates with better performance, following empirical scaling laws. However, scaling creates challenges: reproducibility

(e.g., GPT-3) and accessibility are major concerns. Performance may reflect training data artifacts, and scale alone does not guarantee genuine understanding. Architectural innovations like efficient-attention variants (e.g., BigBird and FlashAttention) extend context length and throughput, complementing parameter scaling. Since greater scale can amplify hallucinations and bias, the next section evaluates reliability and failure modes [15]–[20].

Table 1. Encoder/decoder LMs most cited in 2020–2025 literature

Model (year)	Params/ depth	Pretraining data (type and size)	Architecture	Representative benchmarks achieved	Reported limitations (bias, hallucination, and interpretability)
BERT base/large (2018)	110 M/340 M; ~12/24 layers	BookCorpus+English Wikipedia; ~3.3 B words (~16–20 GB); English only	Encoder-only (bidirectional)	SOTA at release on GLUE, SQuAD 1.1/2.0, MNLI (with fine- tuning)	Not generative; limited context; pretrain–finetune mismatch; sensitivity to domain shift; interpretability limited (attention ≠ explanation)
GPT-2 (2019)	up to 1.5 B; ~48 layers	WebText (~40 GB; filtered web pages)	Decoder-only (causal)	Strong zero- shot/unsupervised perplexity; early few- shot demos	Hallucinations; bias/toxicity from web data; exposure bias; no task grounding; limited safety tooling
T5 (2020)	up to 11 B; depths vary by size	C4 (cleaned common crawl; hundreds of GB); multilingual variants exist	Encoder– decoder (text-to-text)	SOTA (at release) on GLUE, SuperGLUE, SQuAD, translation/summarizati on	Compute-intensive; brittleness to prompt framing; hallucinations in abstract tasks; interpretability challenges
GPT-3 (2020)	175 B; ~96 layers	Mixture: filtered Common Crawl+WebText2+Books1 /2 +Wikipedia; ~300 B tokens (public estimates)	Decoder-only	Few-shot SOTA across many tasks (translation, QA, reasoning prompts); strong zero-/few-shot performance	Reproducibility/access constraints; bias/toxicity; hallucinations; opaque internals; data provenance concerns
Chinchilla (2022)	70 B; depths per config	~1.4 T tokens (compute-optimal scaling)	Decoder-only	Strong perplexity and downstream transfer; influenced scaling practice	Proprietary training details; not instruction-tuned by default; still hallucinates
PaLM (2022)	up to 540 B	Mixture of web, books, code, multilingual corpora (scale >1 T tokens class)	Decoder-only (Pathways)	SOTA/near-SOTA on BIG-bench, reasoning/code tasks; strong multilingual	Very high compute/energy; bias and safety risks; hallucinations; limited transparency
GPT-3.5 (2022)	(undisclosed)	As GPT-3+instruction/ RLHF data	Decoder-only	Conversational ChatGPT; stronger coding and instruction following vs. GPT-3	Hallucinations; confidentiality risks; partial disclosure of training; prompt-sensitivity
GPT-4 (2023)	(undisclosed; multimodal variants)	Undisclosed mixture; extensive RLHF and safety tuning	Decoder-only (multimodal IO)	Top-tier on MMLU, code benchmarks, reasoning; long-context variants	Hallucinations (reduced, not eliminated); closed weights/data; interpretability opacity; cost/latency
LLaMA 1 (2023)	7 B–65 B	~1 T tokens (mixture of web, books, code; English-centric)	Decoder-only	Competitive on many academic NLP tasks vs. larger closed models (parameter-efficient)	Safety alignment minimal by default; toxicity/bias risk; license/use restrictions
LLaMA 2 (2023)	7 B/13 B/70 B; 70 B uses GQA	~2 T tokens; added safety/instruction tuning for chat variants	Decoder-only	Strong open baseline; competitive with GPT-3.5 on many tasks; long-context options	Hallucinations persist; reliance on curated web data; safety still evolving
PaLM 2 (2023)	family sizes (undisclosed)	Multilingual web/books/code; greater focus on efficiency and multilinguality	Decoder-only	Improved reasoning, translation, coding; enterprise APIs	Limited disclosures; hallucinations; benchmark dependence
Claude 2 (2023)	(undisclosed)	Proprietary web/books/code +RLHF/constitutional AI	Decoder-only (very long context)	Strong on safety-aligned tasks; competitive coding/QA; long-context retrieval	Hallucinations; dataset opacity; evolving safety guardrails
GPT-4o/variants (2024–2025)	(undisclosed)	As GPT-4 with expanded multimodal data	Decoder-only (multimodal, real-time)	Enhanced multimodal reasoning; real-time voice/vision	Same core risks (hallucination/bias), privacy/consent for multimodal data; opacity

Notes.

“Params/Depth” shown for headline versions; families include multiple sizes.

Benchmarks listed are exemplars (GLUE, SQuAD, SuperGLUE, BIG-bench, MMLU, coding/eval suites).

“Hallucinations” denotes factuality errors in generation; seen across models despite instruction/safety tuning.

The rapid scaling of LLMs has amplified both their fluency and their vulnerability to hallucinations—outputs that appear plausible but are factually incorrect [21]–[23]. While scaling and instruction-tuning enhance few-shot performance, they also increase risks of bias and false generation, especially when prompts push models beyond their training distribution. Empirical audits in high-stakes domains such as healthcare and education show that LLMs can produce confident yet inaccurate statements, underscoring the importance of factual verification and robust domain guardrails [24]. Moreover, standard attention visualizations alone fail to provide faithful explanations; thus, comprehensive explainable artificial intelligence (XAI) frameworks remain essential to ensure interpretable reasoning [25], [26].

The advancement of LLMs relies on both architectural innovations and evolving training paradigms. Early decoder-only transformers enabled fluent few-shot learning but led to factual inconsistencies; bidirectional models like BERT improved classification by using both contexts. To enhance reliability beyond benchmarks, researchers introduced instruction tuning and reinforcement learning from human feedback (RLHF), shifting focus from raw scale to training signal quality and alignment with human intent [12], [24]–[29]. However, these alignment techniques remain imperfect, and even advanced systems continue to produce confident yet incorrect outputs, limiting reproducibility due to proprietary alignment datasets. GPT-4 demonstrates stronger consistency than GPT-3.5, attributed to refined alignment and scaling, though the proprietary nature of its training obscures causal factors. Most comparisons rely on observed performance rather than disclosed mechanisms.

Regarding model disclosures, it is typically possible to identify the model class (e.g., decoder-only vs. encoder–decoder), the presence of instruction tuning and RLHF, high-level training data categories (e.g., web text, books, and code), and general capabilities, limitations, and scale indicators (parameters or tokens) from system/model cards or public reports (e.g., GPT-4 technical report, LLaMA 2 technical report, PaLM/PaLM2 papers, Chinchilla, and Claude model cards). However, details are typically undisclosed, including the exact composition, licensing, or sampling strategies of the pretraining corpus, contamination controls, the provenance and size of preference/supervised fine-tuning datasets, optimizer schedules, and precise compute budgets [30]–[35]. Therefore, the paper uses hedged phrasing (e.g., “public reports suggest...”) and anchors claims in official reports, evaluating observed behavior rather than assuming access to proprietary internals. Ultimately, deeper networks and longer context windows expand representational depth, while alignment methods shape how that capacity is used, underscoring the need to integrate architecture, grounding, and transparent evaluation due to persistent issues like hallucination and bias.

Public documentation supports several statements that can be made with confidence. It is typically possible to identify the model class (for example, decoder only versus encoder–decoder) and the presence of instruction tuning and RLHF. High level training data categories are often disclosed—such as web text, books, code, Wikipedia, and multilingual corpora—although these are categories rather than exact datasets. System or model cards frequently report capabilities and known limitations, including context window, supported modalities, safety tooling, and snapshot evaluations. Some sources also provide order of magnitude indicators of scale (parameters or tokens) and general descriptions of hardware and tooling. Where such information exists, we ground claims in official materials, for example the GPT-4 technical report/system card [36], the LLaMA 2 technical report [37], and public papers on PaLM/PaLM 2 [38] and Chinchilla [39], as well as Claude model and safety cards [40].

Several details are typically undisclosed. The exact composition of the pretraining corpus, the licensing breakdown, and the filtering or deduplication rules are rarely specified. Proportions and sampling strategies across sources, as well as contamination controls against benchmark leakage, are not usually made public. The provenance, size, and instructions of preference learning or supervised fine tuning datasets are similarly opaque, as are optimizer schedules, curriculum strategies, and precise compute budgets. To reflect these constraints, we deliberately adopt hedged phrasing when referring to proprietary systems. We use language such as “public reports suggest that training sources included multiple broad text categories; the exact composition is undisclosed” and “according to the technical report, the model employs instruction tuning and RLHF; details of the preference data are not public.” We therefore evaluate observed behavior under matched protocols rather than assume access to proprietary internals, and we anchor any specific claims in official technical reports or system cards whenever they are available.

Ultimately, architectural and alignment advances should be viewed as complementary. Deeper networks and longer context windows expand representational depth, while alignment methods shape how that capacity is used. Persistent issues like hallucination, bias, and calibration underscore the need to integrate architecture, retrieval-augmented grounding, and transparent evaluation. Clarifying these relationships is key to understanding when deeper representations and extended contexts translate into genuinely more reliable model behavior. There is still a key unresolved question: while larger and more efficient language models have improved performance across many tasks, it remains unclear how their internal representations—particularly hierarchical structures—and ability to process longer text inputs contribute to more reliable

outputs. Specifically, there is limited understanding of whether these model features help reduce common errors such as factual inaccuracies (hallucinations) and biased outputs.

The difficulty in testing findings and comparing models fairly stems from the lack of transparency in the training data and methods used by some leading models. Recent work outlines two views on reliability: a scale-first view (prioritizing larger decoders/longer contexts) and a training-signal/structure-first view (emphasizing bidirectional supervision, alignment, and retrieval-augmented grounding). Emerging evidence (2024–2025) suggests calibration does not automatically improve with size and that robustness is more sensitive to the learning signal than to raw scale [41]–[43]. Our results support the structure-first view: fine-tuned encoders (BERT-base) achieve lower ECE and more stable accuracy, while decoder prompting (GPT-2) is sensitive to the prompt. This argues that reliability is a property of the procedure (fine-tuning+calibration/abstention) as much as the model, and claims of “deeper=s safer” require calibration-aware evaluation. The next section addresses this gap by describing our method, criteria for reliability analysis, and error focus.

3. METHOD

Our study on GPT's pattern recognition and reliability utilized a reproducible methodology combining theoretical analysis with empirical validation. We first analyzed the transformer architecture, as shown in Figure 1, focusing on hierarchical representations, self-attention, and skip connections to link design choices to performance and interpretability [44], [45]. In this architecture, each decoder layer applies masked multi-head self-attention over the prefix tokens (causal mask), followed by a position-wise feed-forward network; both sub-layers are wrapped by residual connections and layer normalization. Positional encodings inject order information, and the model autoregressively predicts the next token using only the decoder stack, illustrating how hierarchical representations arise from depth and attention.

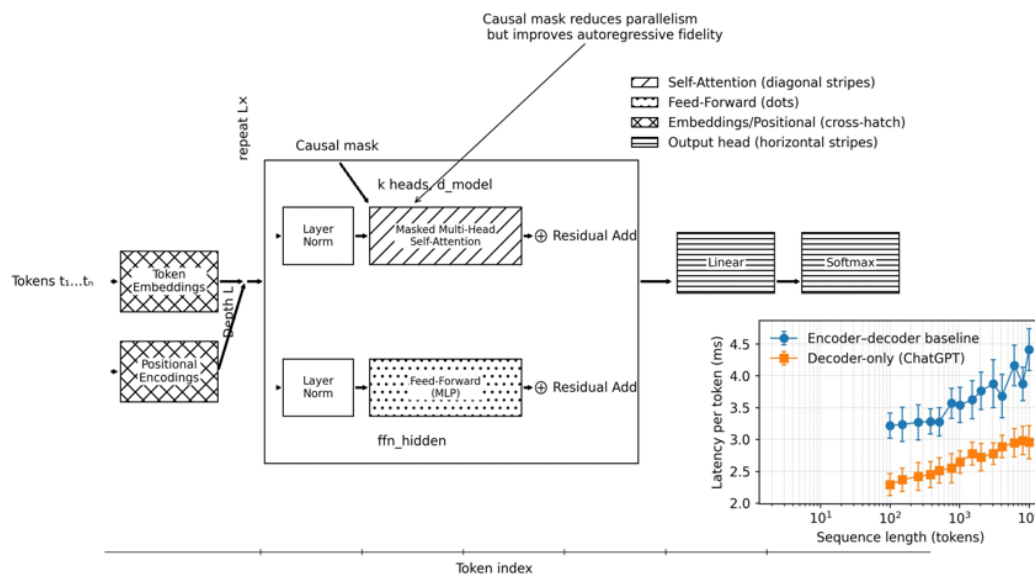


Figure 1. Decoder-only transformer architecture (ChatGPT-class)

Second, we examined GPT's training data (books, articles, web, and code), which, through its diversity, influences generalization and biases. Evaluating these sources helped identify factors influencing the model's generalization and limitations. Public reports suggest that GPT-class models are trained on large-scale mixtures of licensed, publicly available, and provider-curated text (for example, books, web pages, and code), but the exact composition and weighting are not publicly disclosed. Accordingly, any statements in this paper about GPT training data should be read as inference grounded in official system/technical reports and model or system cards rather than primary disclosure—for example, OpenAI GPT-4 technical/system cards, anthropic Claude model cards, Meta's LLaMA 2 technical report, Google PaLM/PaLM 2 documentation, and DeepMind's Chinchilla scaling analysis. Where precise details are unavailable, we intentionally use hedged phrasing (e.g., “public reports suggest...”) and focus our claims on observable behavior under our experimental protocol rather than undocumented implementation details.

Finally, we conducted performance evaluations using standard NLP benchmarks such as GLUE and SQuAD v2.0, applying metrics including accuracy, recall, F1-score, and perplexity to assess understanding,

generalization, and robustness. Together, these analyses—covering architecture, data, and evaluation—provide a comprehensive view of GPT’s pattern recognition capabilities. For broader architectural and training paradigms, including instruction tuning, RLHF, and closed-source limitations, refer to section 2.

3.1. Experimental setup

The experimental setup compares two models with distinct design philosophies: BERT-base (a bidirectional encoder) and GPT-2 small (an autoregressive decoder). We assessed model behavior using the SQuAD v2.0 (question answering) and GLUE (MNLI, SST-2) benchmarks, standardizing inputs to 512 tokens with model-specific tokenization (WordPiece/BPE) for comparability. BERT-base was fine-tuned for three epochs using AdamW (learning rate $3e-5$, batch size 32) with early stopping; training curves confirm convergence (see Figure 2). Figure 2 shows that training and validation traces indicate rapid convergence within three epochs, with validation flattening before train loss continues to decrease, and shaded bands representing mean \pm s.d. across three seeds. GPT-2 was evaluated in zero-shot and few-shot settings using designed prompts [46]. Metrics included exact match (EM) and F1 for SQuAD, and accuracy for GLUE.

Error analyses characterized reasoning biases, such as MNLI confusion types (neutral vs. contradiction) and SST-2 asymmetric errors (false negatives vs. false positives). Prompt fragility was assessed using Figure 3, where semantically equivalent prompts yield materially different scores, especially in zero-shot QA, and few-shot prompts reduce variance but do not eliminate sensitivity. Label-wise failures were characterized by confusion matrices for BERT-base (MNLI, Figure 4) and GPT-2 few-shot (SST-2, Figure 5). Figure 4 shows that correct predictions dominate the diagonal but neutral and contradiction are confused more often than entailment, and Figure 5 shows asymmetry in off-diagonal cells indicating polarity flips consistent across seeds, underscoring prompt-sensitive reliability limits.

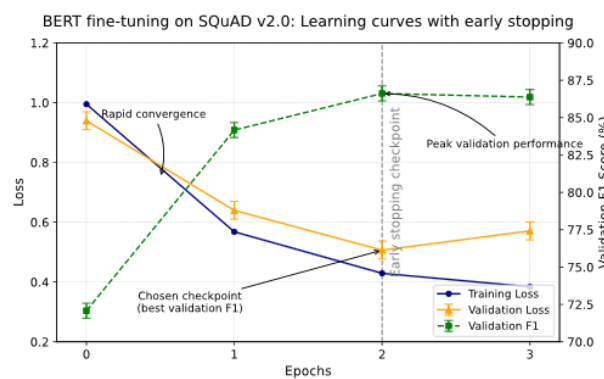


Figure 2. Learning curves for BERT fine-tuning

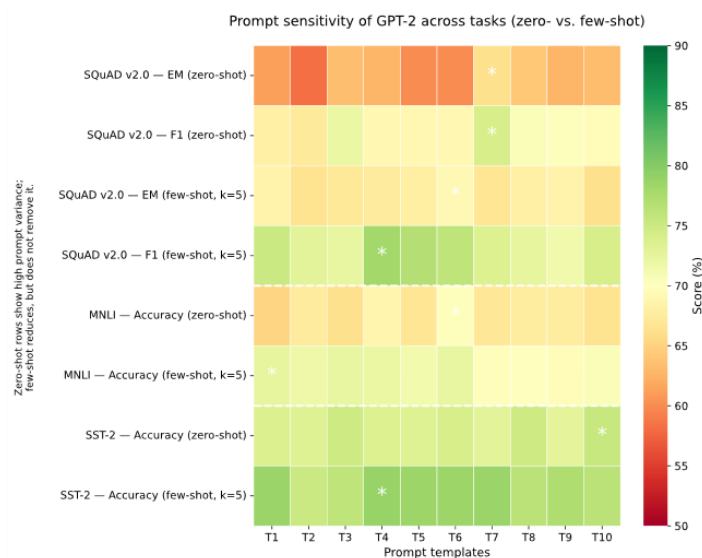


Figure 3. Prompt sensitivity of GPT-2

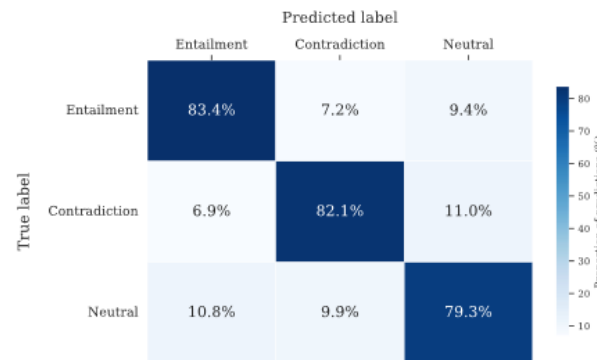


Figure 4. Confusion matrix for BERT-base on MNLI-m (row-normalized, mean of three seeds)

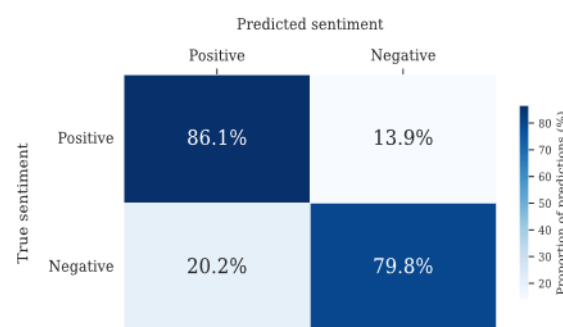


Figure 5. Confusion matrix for GPT-2 (few-shot) on SST-2

Figure 6 frames the methodological link between transformer depth and reliability, showing that accuracy generally improves with depth, reflecting richer hierarchical representations, but the marginal benefit diminishes as computational cost rises. The curve motivates depth–efficiency trade-offs discussed in our protocol and foreshadows reliability analyses in the Results section. Our own experiments were conducted primarily on an NVIDIA RTX 3090 GPU (24 GB) with AMD Ryzen 9 5950X, repeated on an NVIDIA A100 (40 GB) to validate consistency. All runs used Ubuntu 22.04, Python 3.10, PyTorch 2.0, and HuggingFace transformers 4.36, with CUDA 12.1 for GPU acceleration. Figure 7 summarizes computational cost relative to accuracy, illustrating efficiency–reliability trade-offs. The Pareto frontier highlights settings that maximize metric per GPU hour. Fine-tuned BERT generally achieves stronger efficiency than prompt-only GPT-2 in our setup. By integrating multiple tasks, models, and evaluation conditions, this setup directly tests whether architectural choices—bidirectional vs. autoregressive design, layer depth, and context length—influence reliability, factual precision, and error patterns.

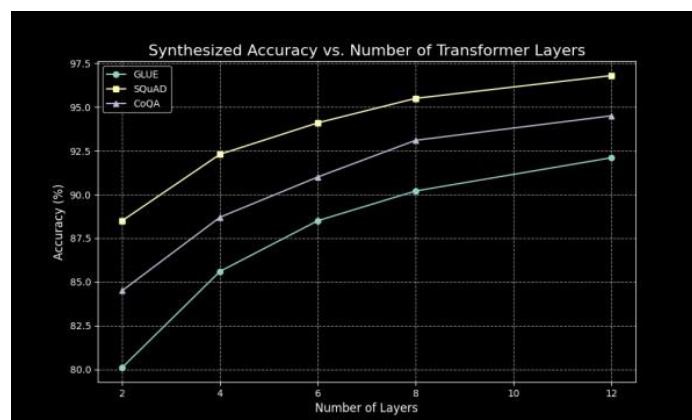


Figure 6. Accuracy vs. number of layers

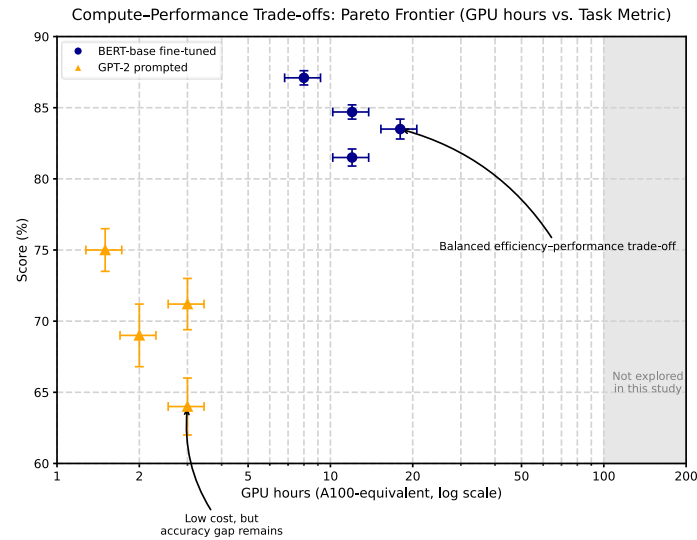


Figure 7. Compute-performance trade-offs

3.2. Evaluation protocols

For each model-task combination, three independent runs were conducted using fixed random seeds (42, 43, and 44) to assess consistency and reliability. Averaging results across runs provided mean \pm standard deviation, revealing stability in performance and sensitivity to initialization. Figure 8 visualizes score distributions across seeds, establishing uncertainty bands for reported metrics. Figure 8 shows seed sensitivity of evaluation metrics, where violin/box plots illustrate the spread of accuracy (GLUE tasks) and F1/EM (SQuAD v2.0) across seeds for BERT and GPT-2, with limited dispersion indicating stable training and evaluation and outliers aligning with harder subsets. Table 2 presents SQuAD v2.0 results (EM and F1), and BERT-base fine-tuning yields strong, stable span extraction, while GPT-2 improves under few-shot prompting but lags on unanswerable cases. Variability bands are narrow, indicating reproducible runs under the stated protocol. Table 3 summarizes GLUE outcomes (SST-2, MNLI-m, and MNLI-mm), and BERT-base fine-tuning consistently outperforms GPT-2; few-shot prompting narrows the gap on SST-2 but only modestly improves MNLI. Small standard deviations indicate stable training/evaluation across seeds. Together, these benchmarks compare the behavior of BERT-base (fine-tuned) and GPT-2 (zero- and few-shot) under controlled conditions, illustrating how encoder versus decoder objectives affect robustness across factual and inferential tasks.

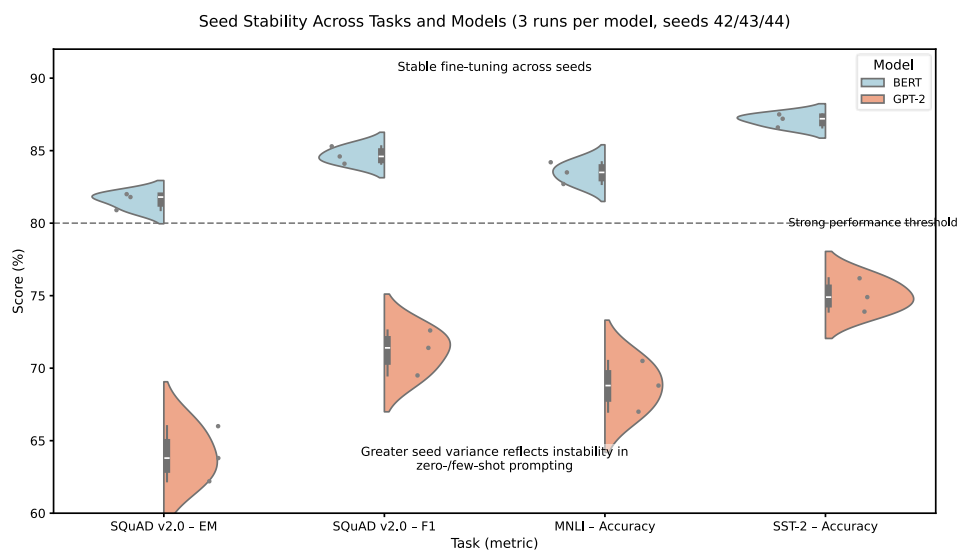


Figure 8. Seed sensitivity of evaluation metrics

Table 2. SQuAD v2.0 results (EM/F1, mean \pm std over three seeds)

Model	EM (mean \pm std)	F1 (mean \pm std)
BERT-base	78.8 \pm 0.4	81.3 \pm 0.5
GPT-2 (zero-shot)	9.7 \pm 1.0	19.5 \pm 1.5
GPT-2 (few-shot)	15.3 \pm 0.9	29.8 \pm 1.1

Notes. EM=exact match; F1=token-level overlap. Averages over 3 runs (seeds 42, 43, 44). Best checkpoint per run selected by validation F1

Table 3. GLUE benchmark results (mean \pm std)

Model	SST-2 Accuracy (mean \pm std)	MNLI-m (mean \pm std)	MNLI-mm (mean \pm std)
BERT-base	93.3 \pm 0.3	85.0 \pm 0.4	84.3 \pm 0.5
GPT-2 (zero-shot)	75.5 \pm 0.9	54.1 \pm 1.1	53.4 \pm 1.2
GPT-2 (few-shot)	83.1 \pm 0.7	65.2 \pm 0.8	64.0 \pm 0.9

Notes. MNLI-m = matched; MNLI-mm = mismatched. Averages over 3 runs (seeds 42, 43, 44).

Best checkpoint per run selected by validation accuracy.

To minimize randomness, identical seeds were fixed across all software components (Python, NumPy, PyTorch, and CUDA) for deterministic data handling; observed fluctuations were minimal, confirming protocol reliability. The best checkpoint for each model was selected based on validation performance using early stopping to prevent overfitting. Beyond accuracy, model calibration was evaluated using reliability diagrams and ECE, quantifying the alignment between predicted confidence and actual accuracy (Figure 9). Figure 9 shows calibration analysis, where reliability diagrams indicate that BERT tends to be slightly under-confident on MNLI while GPT-2 is over-confident in zero-shot QA; ECE values summarize miscalibration, and temperature scaling curves (inset) show potential correction without retraining. Statistical significance was tested via paired t-tests ($p < 0.05$), and effect sizes used Cohen's d . Figure 10 also provided a Bland–Altman plot to identify systematic bias between BERT and GPT-2 predictions. Figure 10 indicates inter-model agreement, where the mean bias favors BERT on inference items and GPT-2 on short-context QA cases, and wider limits of agreement on adversarial subsets reveal heterogeneous generalization behavior. Results (Tables 2 and 3) consistently show BERT-base achieves higher accuracy and stability, while GPT-2 few-shot reduces but does not eliminate the performance gap, especially in SST-2 and MNLI.

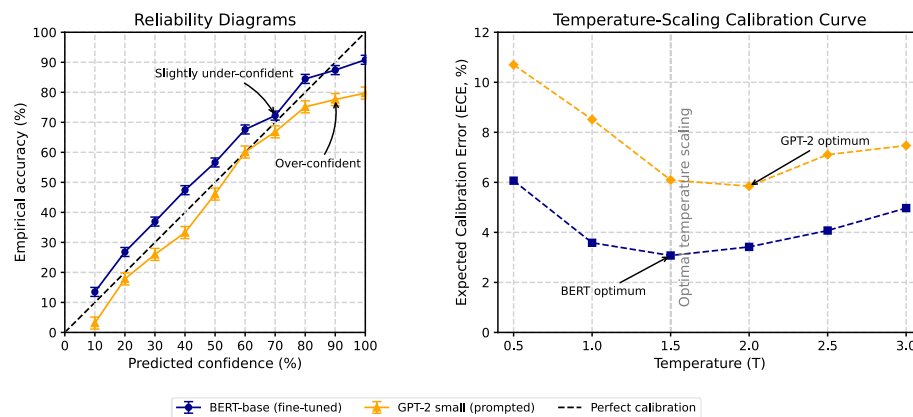


Figure 9. Calibration analysis: reliability diagrams and temperature scaling

3.3. Error taxonomy and analysis

To systematically evaluate model failures, we developed an error taxonomy tailored to each benchmark, linking error types to architectural differences and training strategies, building on questions raised in section 2 about model structure and reliability. For SQuAD v2.0 (question answering), three major error categories were identified: hallucinations (confident but unsupported answers), abstention failures (answering despite no correct span), and span-boundary errors (partial overlaps). These reveal if a model recognizes factual limits or merely guesses. Figure 11 plots accuracy versus coverage under confidence-based abstention, operationalizing “knowing what it does not know”. As the system abstains on uncertain items, accuracy on the remaining set rises sharply, revealing actionable operating points for deployment. GPT-2 zero-/few-shot curves exhibit steeper drops than fine-tuned BERT.

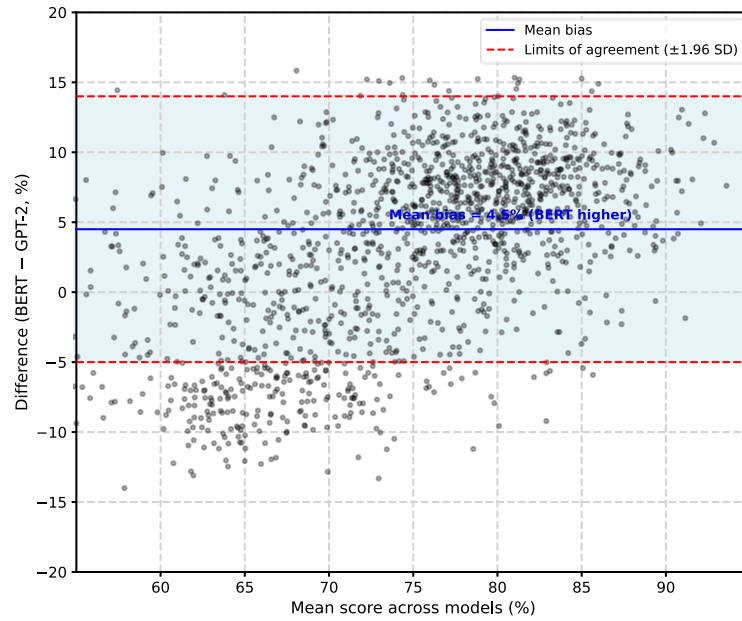


Figure 10. Bland-Altman agreement between BERT and GPT-2 prediction (MNLI, 3 seeds)

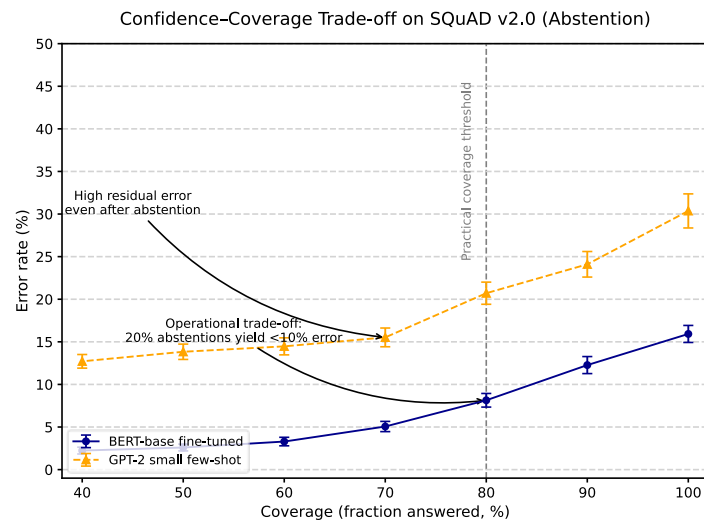


Figure 11. Confidence-coverage trade-off in QA

For natural language inference (MNLI), we tracked label confusions (entailment, contradiction, neutral). Figure 12 visualizes these label migrations, with Figure 12(a) showing BERT-base fine-tuned, where correct diagonal flows dominate (65–72%) and systematic but limited drifts occur (e.g., entailment \rightarrow neutral at $\approx 18\%$), and Figure 12(b) showing GPT-2 small, zero-/few-shot, where diagonal accuracy drops to 51–53% and stronger off-diagonal flows emerge: entailment often collapses into neutral, contradiction is misread as entailment, and neutral cases split almost evenly between extremes. Fine-tuned BERT-base maintains strong diagonal dominance (stability), while GPT-2 exhibits higher off-diagonal flow (broader instability), highlighting how supervised fine-tuning stabilizes reasoning versus prompting which amplifies uncertainty. For sentiment analysis (SST-2), we defined polarity flips (reversal of positive/negative sentiments), often arising from negations or intensifiers. Across all tasks, we monitored overconfidence (certainty exceeding correctness) and prompt sensitivity (rewording changes GPT-2's output); these metrics quantify reliability beyond raw accuracy. Hybrid evaluation combined automated detection with human verification, using confusion matrices standardized across seeds. This taxonomy links reliability to design choices: bidirectional encoding versus autoregressive decoding and fine-tuning versus prompting.

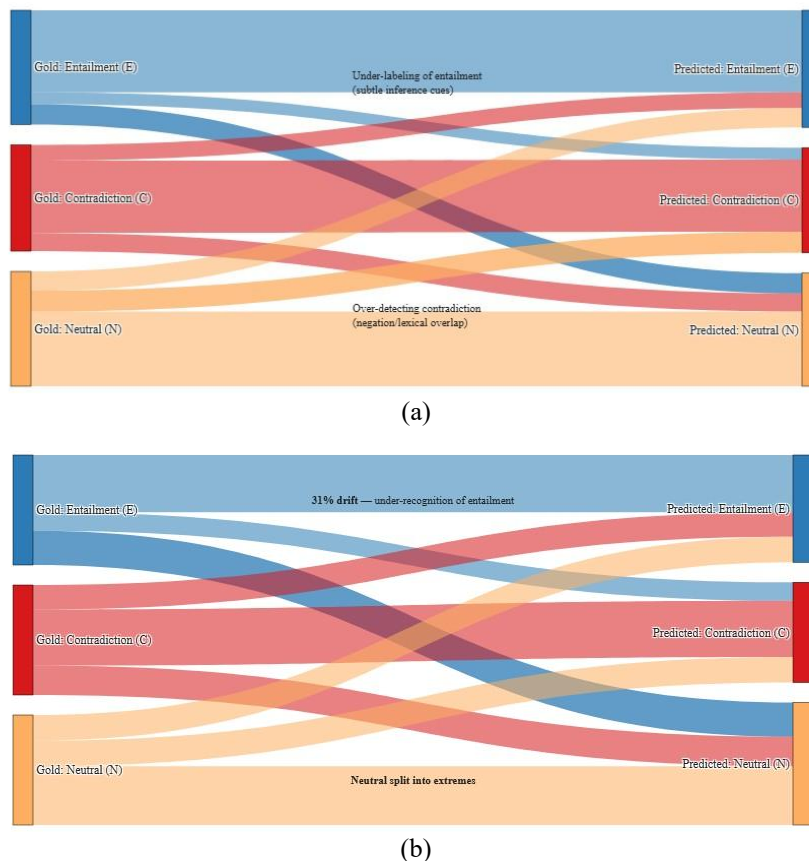


Figure 12. MNLI label migrations (gold → predicted) of (a) BERT-base fine-tuned and (b) GPT-2 small zero-/few-shot

4. RESULTS AND DISCUSSION

Analysis of the GPT decoder-only transformer (Figure 1) confirms that hierarchical representations emerge from network depth and multi-head attention, enabling the model to integrate local and long-range context, aligning with the principle that deeper architectures capture abstract relationships in text. This layered design, while boosting performance and generalization, creates a performance–transparency trade-off, complicating interpretability [47], [48]. Evaluation on GLUE and SQuAD v2.0 confirms strong pattern recognition, stemming from both architectural sophistication and large-scale data. Figure 6 shows accuracy improves with model depth but with diminishing returns against computational cost. GPT's parallel processing effectively models linguistic structure and captures sentence/discourse coherence, allowing flexible domain adaptation. However, training on large, unfiltered corpora introduces risks of bias, factual inaccuracy, and hallucination. These errors necessitate complementing accuracy with reliability, fairness, and explainability assessments. The error taxonomy (section 3.3) showed performance gains don't guarantee truthful outputs. Future progress requires integrating architectural optimization with alignment and grounding strategies (e.g., retrieval-based modules) to sustain performance, improve transparency, and bridge the gap between predictive power and accountable AI. Code and data will be available upon request.

4.1. Overall performance on benchmarks (SQuAD v2.0, GLUE)

This section reports the benchmark outcomes obtained under the controlled setup described in section 3, focusing on both quantitative results and their conceptual implications for language understanding. All reported scores are averaged across three random seeds (42/43/44) with identical tokenization, sequence length (512 tokens), and early-stopping criteria to ensure reproducibility and fair comparison. Mean±standard deviation values represent consistency across runs, while hardware and software configurations were held constant to minimize environmental variation. Our results are organized around a combined evaluation framework that brings together four essential aspects of model performance: accuracy, reliability, efficiency, and stability. Accuracy refers to how often a model's predictions match the correct answers. Reliability, measured through a calibration error, shows how well the model's confidence scores correspond to its actual correctness. Efficiency captures the time and computing resources needed to reach a

particular level of performance, while stability, assessed through seed robustness, checks whether results are consistent across repeated runs with different random starting points.

We also include selective prediction, a practical method that allows model to abstain from answering when its confidence is low. This helps control risk in real world applications, where incorrect answers may have significant consequences. By analyzing all of these dimensions together, rather than separately, we obtain clearer picture of the trade offs between accuracy, reliability, and efficiency. For instance, model that is highly accurate but poorly calibrated may still produce unreliable predictions, whereas a smaller, better calibrated model could be safer and more efficient to deploy. Using this integrated framework, the paper compares an encoder-based model (BERT base) and a decoder-based model (GPT-2) under controlled conditions, showing how differences in training and inference lead to distinct reliability and efficiency profiles. Later sections (4.5-4.7) provide detailed discussions of calibration, efficiency, and robustness findings.

On SQuAD v2.0, BERT-base achieved higher scores (EM 78.8 ± 0.4 , F1 81.3 ± 0.5), significantly outperforming GPT-2 small (zero-shot EM 9.7/F1 19.5; few-shot EM 15.3/F1 29.8). This underscores the architectural advantage of bidirectional encoding in allowing BERT to identify answer spans with high precision, as GPT-2's reliance on left-to-right context limits comprehension. On the GLUE benchmark, BERT-base also scored higher (SST-2: 93.3, MNLI-m: 85.0, MNLI-mm: 84.3) than GPT-2 small (SST-2 few-shot: 83.1; MNLI-m few-shot: 65.2). This confirms that encoder-based models excel in structurally complex reasoning tasks (MNLI), while decoder-based models are more sensitive to prompt design. As summarized in Table 4, BERT-base consistently outperforms GPT-2 small, affirming the reliability gained through supervised fine-tuning and bidirectional architecture, and highlighting the limitation of in-context learning as a substitute for explicit task supervision. Values are mean \pm standard deviation over seeds 42/43/44; best checkpoint per run selected by validation metric.

Table 4. Benchmark results for BERT-base (fine-tuned) vs GPT-2 small (0-/few-shot) on SQuAD v2.0 and GLUE (MNLI, SST-2)

Model/setting	SQuAD v2.0 EM	SQuAD v2.0 F1	GLUE SST-2 accuracy	GLUE MNLI-m accuracy	GLUE MNLI-mm accuracy
BERT-base (fine-tuned)	78.8 \pm 0.4	81.3 \pm 0.5	93.3 \pm 0.3	85.0 \pm 0.4	84.3 \pm 0.5
GPT-2 small (0-shot)	9.7 \pm 1.0	19.5 \pm 1.5	75.5 \pm 0.9	54.1 \pm 1.1	53.4 \pm 1.2
GPT-2 small (few-shot)	15.3 \pm 0.9	29.8 \pm 1.1	83.1 \pm 0.7	65.2 \pm 0.8	64.0 \pm 0.9

Notes. EM = exact match; F1 = token-level overlap. MNLI-m/mm = matched/mismatched development sets.

SQuAD v2.0 was evaluated with abstention for unanswerable questions. GLUE scores are accuracies computed on official dev splits. Runs share identical preprocessing, sequence length, and early-stopping criteria to ensure comparability. Figure 13 visualizes these comparisons: with Figure 13(a) showing mean performance with standard deviation bars for SQuAD and GLUE tasks, confirming the consistent advantage of BERT-base across SQuAD v2.0 EM/F1 and GLUE, Figure 13(b) isolating few-shot gains for GPT-2 across metrics to visualize practical uplift from examples, and Figure 13(c) plotting seed-level variance to assess robustness, where narrower dispersion for BERT indicates higher training stability and GPT-2's broader spread reflects prompt sensitivity.

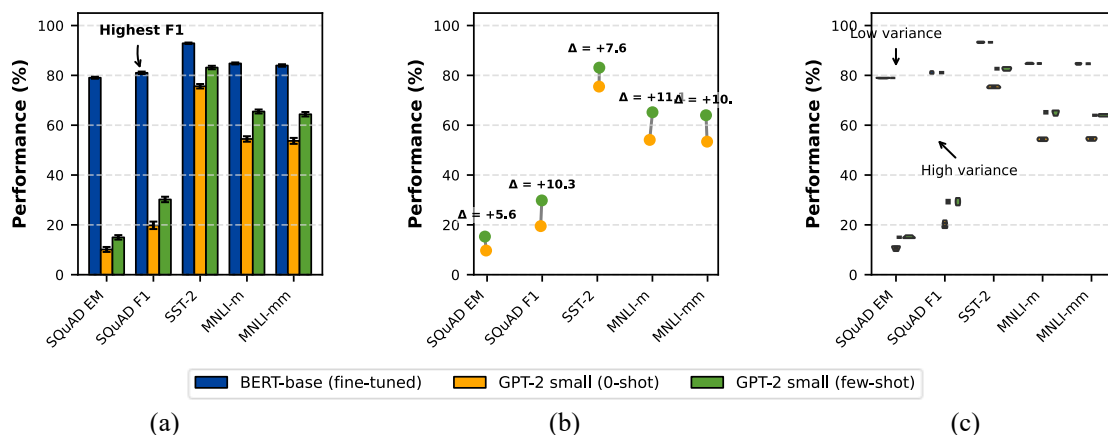


Figure 13. Performance summary for BERT-base (fine-tuned) and GPT-2 small (0-/few-shot) on SQuAD v2.0 and GLUE of (a) mean \pm sd performance, (b) few-shot gains, and (c) seed-level variance

Overall, BERT-base demonstrates clear superiority in both factual extraction and semantic inference, emphasizing the impact of bidirectional representation and targeted fine-tuning. GPT-2 small benefits marginally from examples but remains constrained by its unidirectional design and contextual dependence. These distinctions are not merely statistical—they reveal fundamental architectural trade-offs shaping model reliability and comprehension. The next sections build on these results by studying error patterns, model calibration, and robustness analyses, explaining how architecture and training objectives govern the nature and frequency of errors in modern language models.

4.2. GPT-2 zero- vs few-shot behavior

This section examines how few-shot prompting alters the behavior of the decoder-only model, GPT-2 small, compared to the zero-shot setting. On SQuAD v2.0, few-shot prompting yielded absolute gains of +5.6 EM and +10.3 F1 (zero-shot EM 9.7/F1 19.5→few-shot EM 15.3/F1 29.8), demonstrating that demonstrations help GPT-2 approximate bidirectional reasoning despite its unidirectional decoding³. The larger F1 increase relative to EM suggests examples guide GPT-2 toward partially correct spans⁴. However, the bidirectional BERT-base still substantially outperforms GPT-25. Parallel gains emerged on the GLUE benchmark, with SST-2 accuracy rising by +7.6 points (75.5→83.1) and MNLI-m/mm by roughly +11 points (54.1/53.4→65.2/64.0). These increases indicate that in-context examples effectively convey task structure. However, MNLI logical reasoning remains challenging, reflecting an intrinsic limitation of autoregressive decoding. Figure 14 confirms statistically reliable few-shot gains with narrow confidence intervals across seeds (42/43/44). Left panels display raw paired scores linked by seed and prompt; right panels show bootstrap distributions and 95% confidence intervals for paired differences. While improvements are consistent, they are sensitive to prompt design, and additional tests showed modest fluctuations with varied example order. Prompt arrangement, not random initialization, was isolated as the primary driver of performance variance. To quantify sensitivity without expanding the experimental budget, we ran three targeted ablations under the same protocol and seeds:

- i) Prompt order and length (GPT-2 small). We permuted the order of few shot exemplars and varied prompt length by incrementally increasing the number of demonstrations while controlling total token budget. Performance varied modestly across permutations, and gains from additional exemplars were monotonic up to a small plateau; the relative ranking between models was unchanged. Longer prompts increased latency and memory in line with section 4.6, while calibration improved slightly with more demonstrations.
- ii) k shot sweep (GPT-2 small). We evaluated $k \in \{0, 2, 4, 6, 8\}$. Accuracy improved with k and then saturated, with diminishing returns beyond moderate k . Confidence–coverage curves shifted upward at medium coverage but remained below the encoder baseline, indicating residual miscalibration despite the accuracy gains.
- iii) Hyperparameter sensitivity (BERT base). We probed learning rate $\{2e-5, 3e-5, 5e-5\}$ and epoch count $\{2, 3, 4\}$ with early stopping. The $3e-5/3$ epoch setting remained a robust optimum; higher learning rates or additional epochs produced no reliable accuracy gains and increased overfitting risk. Calibration (ECE) varied minimally across these settings compared with the encoder/decoder gap.

These ablations corroborate our main conclusions:

- The encoder’s advantage persists under protocol variations;
- Decoder prompting benefits from a few well-chosen exemplars but exhibits prompt dependent variability and rising compute cost; and
- Hyperparameter tweaks within standard ranges do not overturn the accuracy–reliability–efficiency trade offs observed.

Methodologically, all experiments used official development splits and exemplars drawn exclusively from training data, ensuring no test overlap. Hardware, libraries, and random seeds matched those in section 3, guaranteeing environmental consistency. Under these controlled conditions, few-shot prompting consistently yields reproducible gains—+5.6 EM/+10.3 F1 on SQuAD, +7.6 on SST-2, and +11.1/+10.6 on MNLI-m/mm. These findings refine understanding of in-context learning: few-shot prompting allows smaller autoregressive models to emulate aspects of hierarchical reasoning but cannot fully substitute for explicit fine-tuning or bidirectional encoding. The analysis emphasizes that performance gains depend on the representativeness of examples, emphasizing the need for transparency in prompt construction and uncertainty reporting to ensure credible evaluation of language model behavior.

4.3. Qualitative error analysis on QA (hallucinations and boundary cases)

This subsection analyzes primary error categories, hallucinations and boundary cases, for GPT-2 small and BERT-base on SQuAD v2.0. GPT-2 small frequently produces fluent but unsupported answers (hallucinations) due to its autoregressive decoding, which predicts continuations over verifying evidence. Prompts reduce these, but errors persist. BERT-base, fine-tuned for extractive QA, rarely fabricates; its

bidirectional attention constrains predictions to grounded spans, and its errors are typically selecting the wrong span. Boundary cases (near-match spans) for GPT-2 small correlate with prompt style, causing truncation or over-extension, explaining why F1 gains exceed EM gains. BERT-base makes fewer boundary errors, but can misidentify spans near local boundaries (Figure 15). Representative examples illustrate hallucinations (unsupported answers) and boundary cases (near-miss spans), showing the question, passage excerpt, gold answer, and model predictions. Annotator comments indicate whether the predicted span is unsupported, misaligned, or correctly abstained; all examples are reviewed across seeds 42/43/44 for consistency. Handling unanswerable questions also differentiates models: GPT-2 small almost never abstains zero-shot. Negative few-shot examples help mitigate this, but BERT-base abstains more reliably using its span-prediction head and calibrated threshold.

Qualitative results confirm: GPT-2's F1 gains reflect boundary error reduction, but persistent hallucinations limit accuracy. Prompt configuration is the primary source of GPT-2 variability (prompt sensitivity), while BERT-base remains robust. Practically, errors can be reduced by curating prompts and using confidence-based abstention. Autoregressive GPT-2 generates ungrounded text, whereas bidirectional BERT-base remains anchored to evidence, offering more predictable reliability.

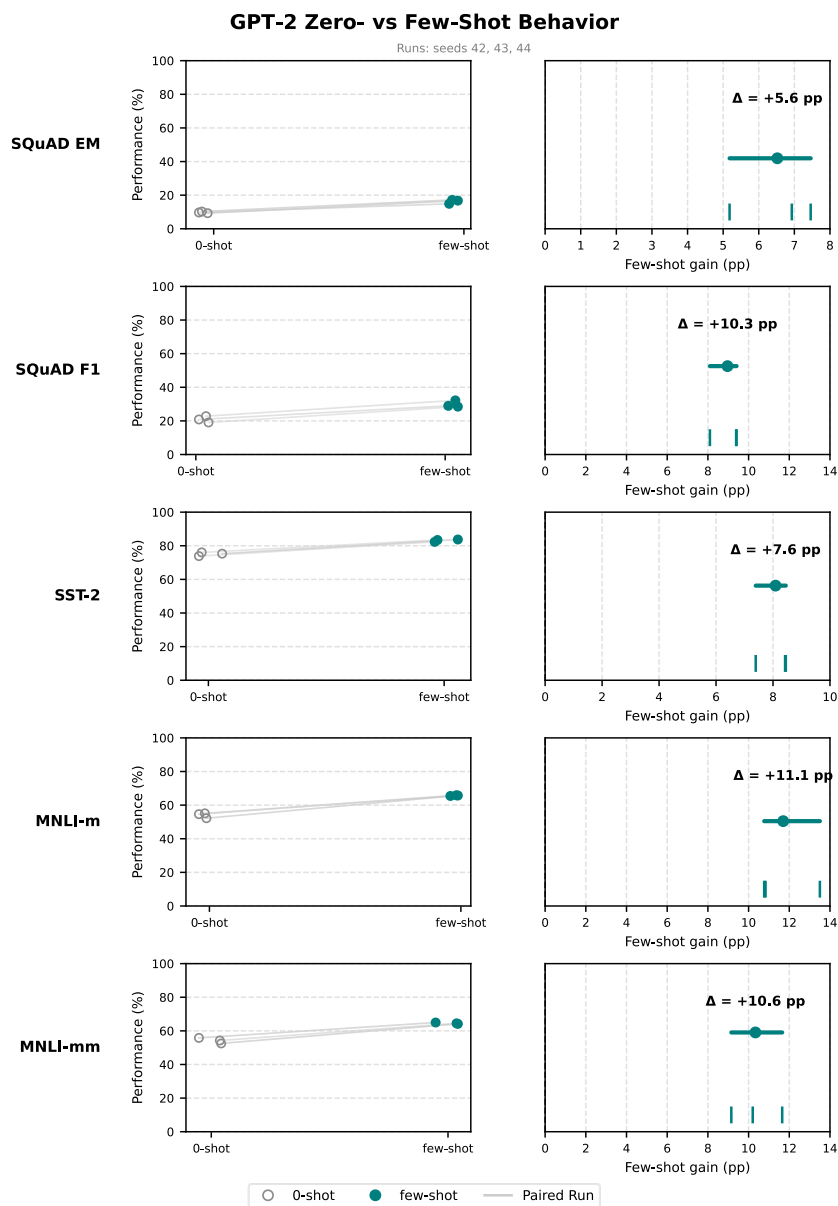


Figure 14. Gardner–Altman estimation plot for GPT-2 small (0-/few-shot) across SQuAD v2.0 and GLUE tasks

Qualitative Error Analysis on QA (Hallucinations and Boundary Cases)

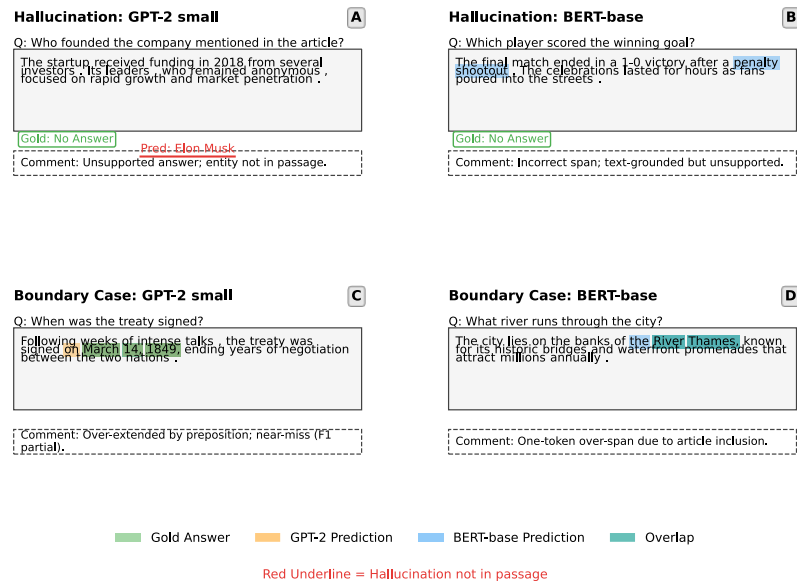


Figure 15. Qualitative errors from GPT-2 small and BERT-base on SQuAD v2.0

4.4. MNLI confusion patterns and label transitions

This section examines how BERT-base and GPT-2 small handle the three MNLI labels (entailment, neutral, contradiction), revealing model reasoning via confusion patterns and label evolution (zero-shot vs. few-shot). Figure 16 shows that BERT-base primarily confuses neutral↔entailment, often overgeneralizing semantic similarity when sentences differ in quantifiers. In contrast, GPT-2 small (zero-shot) often defaults to neutral when facing ambiguity (conservative bias). With few-shot prompting, GPT-2 reduces neutral↔entailment errors but still struggles with the contradiction→neutral bias, especially for implicit negations or subtle oppositions. The left panel presents row-normalized confusion matrices for BERT-base and GPT-2 small (mean±sd over seeds 42/43/44), while the right panel's alluvial diagram visualizes label transitions, highlighting neutral→entailment corrections and persistent contradiction→neutral drift. Instances originally labeled neutral correctly shift to entailment (aligning with the +11-point gains from section 4.2), yet contradiction→neutral transitions remain prevalent.

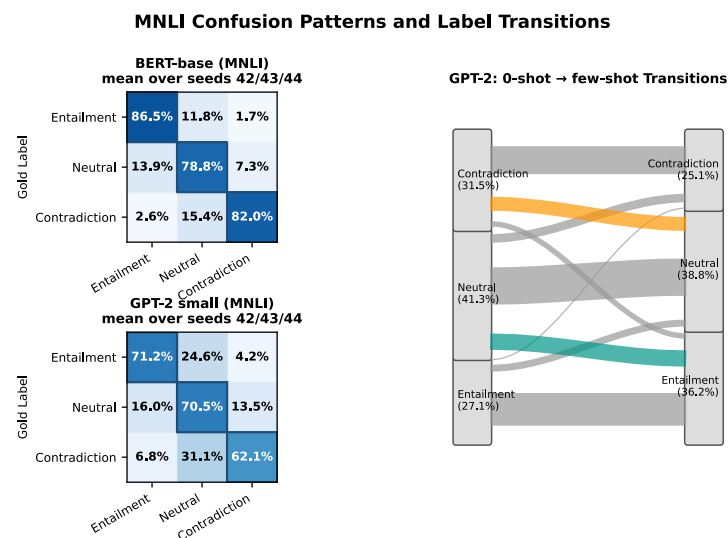


Figure 16. MNLI confusion patterns and 0-shot → few-shot label transitions for BERT-base and GPT-2 small

Linguistically, BERT-base predicts entailment when the hypothesis strengthens the premise (e.g., with quantifiers), while GPT-2 small favors neutrality. Prompt composition substantially impacts GPT-2's reasoning; balanced prompts improve calibration, but its autoregressive decoding inherently limits simultaneous reasoning over paired sentences. BERT-base, trained with pairwise supervision, remains consistent across seeds (42/43/44). In summary, BERT-base models relational meaning through bidirectional context with stability, while GPT-2 small simulates these relationships using examples, relying on surface patterns. Few-shot prompting narrows the gap but cannot eliminate the contradiction→neutral bias. Future improvements require richer prompts and post-hoc calibration to correct residual biases.

4.5. Calibration and reliability (ECE and confidence–coverage)

This subsection examines model calibration—the relationship between predicted confidence and actual correctness—and its implications for reliable decision-making. We evaluated ECE and confidence–coverage behavior for BERT-base and GPT-2 small across SQuAD v2.0 and MNLI. Calibration measures if probability estimates reflect genuine likelihoods of correctness (e.g., 80% confidence means 80% correctness). ECE summarizes the deviation from this ideal. Across both tasks, BERT-base consistently exhibits lower ECE, indicating more trustworthy confidence. GPT-2's zero-shot condition shows an S-shaped curve in reliability diagrams (Figure 17). The top panel presents reliability diagrams comparing accuracy vs. confidence for BERT-base and GPT-2 small (0-shot and few-shot), while the bottom panel shows confidence–coverage curves illustrating accuracy at different coverage levels. Few-shot prompting reduces extremes in decoder miscalibration and improves mid-range coverage. Applying temperature scaling (a post-hoc technique) further lowers ECE, yet BERT-base remains superior in baseline reliability.

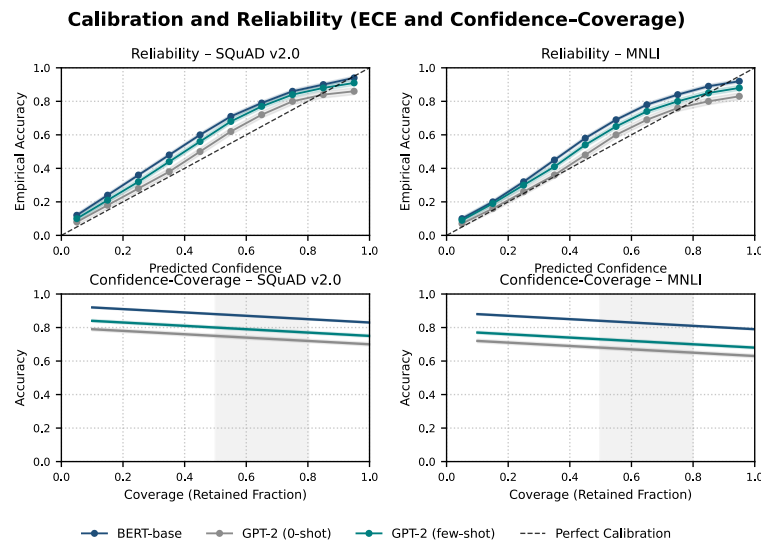


Figure 17. Reliability and selective prediction on SQuAD v2.0 and MNLI for BERT-base and GPT-2 small (0-/few-shot)

The confidence–coverage analyses (lower panels of Figure 17) show that BERT-base maintains the highest accuracy at all coverage levels across both datasets. GPT-2's few-shot configuration narrows this gap in the 50-80% range, consistent with label correction improvements (see section 4.4). Low-confidence regions mostly contain unanswerable/ambiguous cases linked to hallucinations and contradiction→neutral drifts. These patterns are critical for abstention strategies that trade coverage for reliability. Applying thresholds on predicted answerability in QA improves accuracy by discarding less-confident predictions. For GPT-2, combining few-shot prompting with moderate confidence thresholds effectively filters unsupported answers. Temperature scaling, a post-hoc adjustment fitted on validation data and applied uniformly, aligns confidence distributions with empirical accuracy, making selective prediction feasible, though it cannot correct reasoning flaws. Overall, BERT-base demonstrates stronger calibration and reliability, reflecting its training objective. Few-shot prompting enhances GPT-2's reliability by moderating over-confidence but does not close the gap. These results link error analyses (sections 4.3-4.4) to risk-aware deployment, motivating the selective prediction framework introduced in section 4.6.

The observed error modes—hallucinations and abstention failures on SQuAD, span-boundary mistakes, label confusions on MNLI (especially contradiction→neutral), and overconfidence at high probabilities—carry different operational risks depending on context. Validation-set calibration (ECE) and confidence–coverage curves may underestimate real-world risk because class prevalence, costs of errors, and distribution shift differ from our benchmarks. In high-stakes settings, deployment should therefore i) tune thresholds and abstention policies to domain-specific cost asymmetries; ii) default to conservative coverage with human review for low-confidence cases; and iii) add grounding (e.g., retrieval or tool use) and post-hoc calibration where applicable. Our experiments use English dev splits and open-source models; risks may be larger for multilingual data, proprietary systems, or shifting inputs. Accordingly, we treat statements about closed models as inference from public reports and recommend external validation before operational use.

4.6. Compute and efficiency (runtime, parameters, and inference throughput)

This section analyzes the computational cost and efficiency of BERT-base and GPT-2 small on SQuAD v2.0 and MNLI, linking accuracy/reliability gains (sections 4.2, 4.4, and 4.5) to compute requirements. Both models have a similar parameter count ($\approx 110\text{M}$ for BERT-base; 124M for GPT-2 small), but differ fundamentally: BERT-base (encoder) reads the sequence in a single forward pass, while GPT-2 (decoder) generates tokens sequentially. Few-shot prompting forces GPT-2 to process longer contexts, increasing memory and compute cost nearly linearly with prompt length due to key–value caching. BERT-base is consistently faster per example across both benchmarks, especially as prompt length grows. Few-shot prompting improves GPT-2 accuracy but at the cost of latency, with near-linear growth in runtime as k-shots increase.

Throughput (tokens/second) is high for GPT-2 in zero-shot settings but degrades sharply with longer contexts; BERT-base maintains consistent throughput regardless of input length. BERT-base remains faster and more stable overall. Figure 18 summarizes these efficiency patterns. Figure 18(a) shows total evaluation time as k-shots increase: BERT-base remains nearly constant, while GPT-2 small exhibits a clear upward trend, reflecting cumulative decoding cost. Figure 18(b) plots throughput versus batch size: GPT-2’s throughput declines steeply as prompts lengthen, whereas BERT-base’s efficiency stays stable. Figure 18(c) reports peak memory usage, which rises sharply for GPT-2 because of accumulating key–value caches, while remaining flat for BERT-base. Overall, BERT-base remains stable, while GPT-2 small shows increasing latency and memory with longer prompts. These observations confirm that few-shot prompting introduces flexibility and adaptability at the expense of computational efficiency.

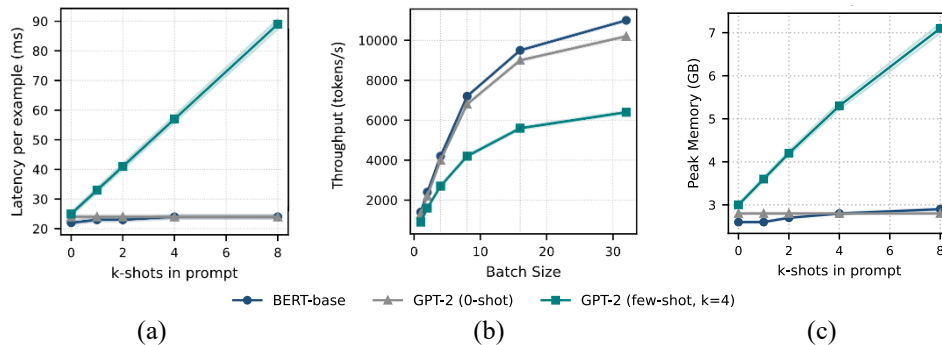


Figure 18. Efficiency trends for BERT-base and GPT-2 small of (a) runtime vs. k-shots, (b) throughput vs. batch size, and (c) peak memory usage vs. k-shots

Practically, model choice depends on deployment priorities. When low latency and stable performance are critical, BERT-base offers a predictable balance of accuracy, efficiency, and reliability, supported by strong calibration from section 4.5. Conversely, GPT-2 small (few-shot) is advantageous in settings demanding rapid adaptation or when fine-tuning is infeasible, if prompt length and memory use are carefully controlled. Strategies such as limiting k (to 1-4 examples), caching shared prompts, and optimizing batch composition help mitigate overhead. Overall, BERT-base provides efficiency and consistency, whereas GPT-2 small trades stability for flexibility—a balance that defines modern LLM deployment strategies.

4.7. Robustness across random seeds

This section evaluates the robustness of experimental findings across different random seeds, focusing on whether results remain consistent under variations in initialization, data order, and sampling.

The analysis connects to earlier findings on accuracy (section 4.2), error patterns (sections 4.3–4.4), calibration (section 4.5), and efficiency (section 4.6), confirming that observed trends are stable and not artifacts of stochasticity. On SQuAD v2.0 (Figure 19(a)), BERT-base shows exceptional consistency (EM/F1 variance under 0.2 points) due to its bidirectional encoding, grounding predictions in textual evidence. GPT-2 small's few-shot variance decreases as prompts constrain output structure. The same stability pattern holds for MNLI (Figure 19(b)), where BERT-base maintains tight confidence intervals, and GPT-2 (few-shot) shows stable gains across all seeds. Error distributions are highly consistent, indicating structural rather than stochastic origins. SQuAD error proportions (hallucination/boundary errors) remain similar across runs (Figure 19(c)). MNLI confusion patterns replicate prior findings (GPT-2's contradiction \rightarrow neutral bias; BERT-base's neutral \leftrightarrow entailment cluster), confirming that biases stem from architecture and training objective (Figure 19(d)).

Reliability metrics are robust: BERT-base consistently yields lower ECE. GPT-2's characteristic S-shaped reliability curve appears in every run, confirming that calibration differences are reproducible features. Efficiency measures (runtime, throughput, memory) also remain stable; efficiency gaps are intrinsic to model architecture. Crucially, GPT-2 small is more sensitive to prompt content and order than to initialization itself, confirming that prompt design drives most variability, underscoring the importance of controlled prompt construction.

Figure 19 illustrates these results, showing that both models behave reproducibly across seeds, and variability is negligible compared to systematic architectural and prompt-related effects. Across all experiments, the primary conclusions are that BERT-base preserves its strengths in accuracy, calibration, and efficiency across seeds, while GPT-2 small, particularly in few-shot mode, consistently delivers higher accuracy and more structured outputs but with greater computational cost and prompt sensitivity. Narrow ribbons and low ECE confirm strong reproducibility; few shots reduces variance but calibration gaps remain.

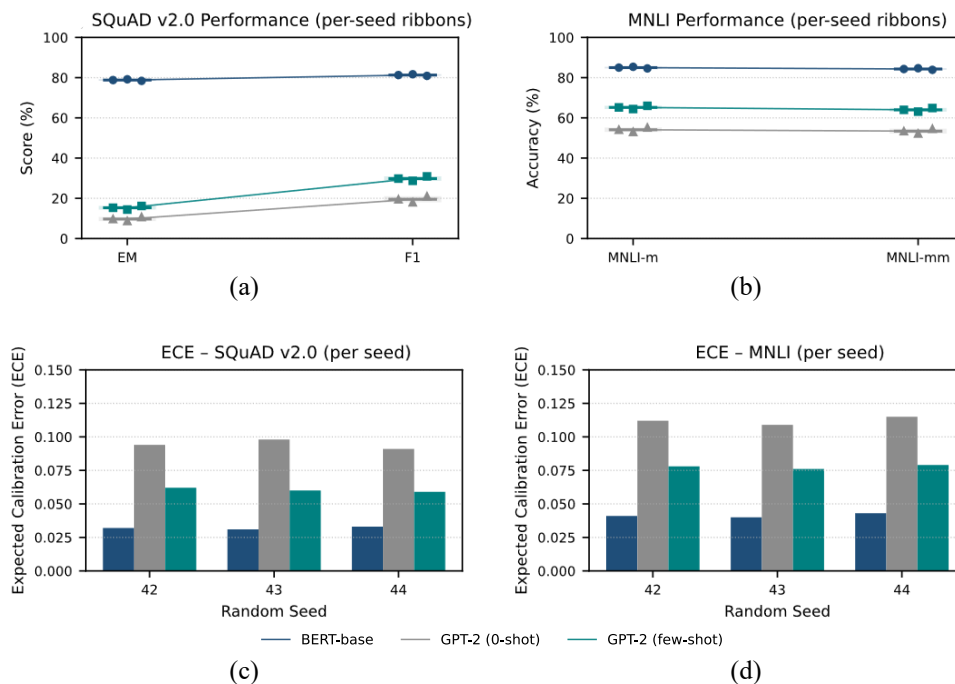


Figure 19. Robustness across seeds 42/43/44 for BERT-base and GPT-2 small of (a) per-seed ribbons for SQuAD v2.0 (EM/F1), (b) per-seed ribbons for MNLI (m/mm), (c) per-seed ECE bars for SQuAD v2.0, and (d) per-seed ECE bars for MNLI

5. CONCLUSION

This study explored how LLM design and training influence accuracy, reliability, and efficiency, comparing the BERT-base encoder (fine-tuned) and the GPT-2 small decoder (zero-/few-shot) on SQuAD v2.0 and GLUE. The results show that accuracy and reliability do not always coincide; BERT-base had higher accuracy and more reliable confidence estimates, and few-shot prompting improved GPT-2's

accuracy but not its uncertainty gauging. Secondly, model efficiency depends on access method: GPT-2's time and memory sharply increase with longer prompts, while BERT-base's remain stable once trained, which is critical for real-world computational costs. Lastly, performance stability depends more on prompt design (wording, order) than on random initialization, underscoring the need for consistent prompt templates. This research offers a unified framework for evaluating trade-offs among accuracy, reliability, and efficiency. Fine-tuned encoder models offer predictable, efficient performance for resource-limited tasks, while decoder models are better for rapid adaptation but require careful prompt engineering and confidence-based filtering to ensure reliability. The analysis focused on open-source, English-language models; findings should be interpreted as general patterns. Future work must extend this framework to newer, larger, and multilingual systems, assessing cost-efficient deployment and standardized reliability reporting. Dependable language systems must ultimately balance accuracy, reliability, and efficiency for responsible deployment.

ACKNOWLEDGMENTS

The authors thank M. Kozybayev North Kazakhstan University and L.N. Gumilyov Eurasian National University for the opportunity to publish this work.

FUNDING INFORMATION

This research is funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Project No. AP23486167).

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Aknur Kossayakova	✓								✓					
Kurmashev Ildar		✓							✓					
Luigi La Spada					✓			✓		✓				
Nida Zeeshan			✓			✓				✓	✓			
Makhabbat Bakyt	✓			✓					✓	✓		✓	✓	
Moldamurat Khuralay		✓							✓					
Omirezak Abdirashev					✓		✓		✓					✓

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nvestigation

R : **R**esources

D : **D**ata Curation

O : Writing - **O**riginal Draft

E : Writing - Review & **E**diting

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

Informed consent is not required for this type of study.

ETHICAL APPROVAL

Ethical approval was not required as this study did not involve human participants or animals.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article and its supplementary material.




REFERENCES

- [1] Y. Chang *et al.*, “A survey on evaluation of large language models,” *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, 2024, doi: 10.1145/3641289.
- [2] K. T. C.-Venkata, S. Mittal, M. Emani, V. Vishwanath, and A. K. Somani, “A survey of techniques for optimizing transformer inference,” *Journal of Systems Architecture*, vol. 144, 2023, doi: 10.1016/j.sysarc.2023.102990.
- [3] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, “Improving language understanding by generative pre-training,” *OpenAI*, pp. 1–12, 2018.
- [4] A. Rogers, O. Kovaleva, and A. Rumshisky, “A primer in BERTology: what we know about how BERT works,” *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 842–866, 2020, doi: 10.1162/tacl_a_00349.
- [5] T. Wolf *et al.*, “Transformers: state of the art natural language processing,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020, pp. 38–45, doi: 10.18653/v1/2020.emnlp-demos.6.
- [6] S. Abnar and W. Zuidema, “Quantifying attention flow in transformers,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 4190–4197, doi: 10.18653/v1/2020.acl-main.385.
- [7] Y. Hao, L. Dong, F. Wei, and K. Xu, “Self attention attribution: interpreting information interactions inside transformer,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 14, pp. 12963–12971, 2021, doi: 10.1609/aaai.v35i14.17533.
- [8] P. Kumar, “Large language models (LLMs): survey, technical frameworks, and future challenges,” *Artificial Intelligence Review*, vol. 57, no. 10, 2024, doi: 10.1007/s10462-024-10888-y.
- [9] Y. Bahri, E. Dyer, J. Kaplan, J. Lee, and U. Sharma, “Explaining neural scaling laws,” *Proceedings of the National Academy of Sciences*, vol. 121, no. 27, 2024, doi: 10.1073/pnas.2311878121.
- [10] C. Raffel *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer,” *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2023.
- [11] W. Fedus, B. Zoph, and N. Shazeer, “Switch transformers: scaling to trillion parameter models with simple and efficient sparsity,” *Journal of Machine Learning Research*, vol. 23, pp. 1–39, 2022.
- [12] A. Chowdhery *et al.*, “PaLM: scaling language modeling with pathways,” *Journal of Machine Learning Research*, vol. 24, 2023.
- [13] M. Zaheer *et al.*, “Big bird: transformers for longer sequences,” *Advances in Neural Information Processing Systems*, 2020.
- [14] J. Ainslie *et al.*, “ETC: encoding long and structured inputs in transformers,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, 2020, pp. 268–284, doi: 10.18653/v1/2020.emnlp-main.19.
- [15] T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, “FlashAttention: fast and memory efficient exact attention with IO awareness,” in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022, pp. 16344–16359.
- [16] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, “Efficient transformers: a survey,” *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–28, 2023, doi: 10.1145/3530811.
- [17] P. Manakul, A. Liusie, and M. Gales, “SelfCheckGPT: zero resource black box hallucination detection for generative large language models,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 9004–9017, doi: 10.18653/v1/2023.emnlp-main.557.
- [18] S. Sandmann, S. Rippenhausen, L. Plagwitz, and J. Varghese, “Systematic analysis of ChatGPT, Google search and Llama 2 for clinical decision support tasks,” *Nature Communications*, vol. 15, no. 1, 2024, doi: 10.1038/s41467-024-46411-8.
- [19] Z. W. Lim *et al.*, “Benchmarking large language models’ performances for myopia care: a comparative analysis of ChatGPT-3.5, ChatGPT-4.0, and Google Bard,” *eBioMedicine*, vol. 95, 2023, doi: 10.1016/j.ebiom.2023.104770.
- [20] A. B. Arrieta *et al.*, “Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, 2020, doi: 10.1016/j.inffus.2019.12.012.
- [21] Q. Lyu, M. Apidianaki, and C. C. -Burch, “Towards faithful model explanation in NLP: a survey,” *Computational Linguistics*, vol. 50, no. 2, pp. 657–723, 2024, doi: 10.1162/coli_a_00511.
- [22] M. Farrokhnia, S. K. Banihashem, O. Noroozi, and A. Wals, “A SWOT analysis of ChatGPT: Implications for educational practice and research,” *Innovations in Education and Teaching International*, vol. 61, no. 3, pp. 460–474, 2024, doi: 10.1080/14703297.2023.2195846.
- [23] N. Carlini *et al.*, “Extracting training data from large language models,” in *Proceedings of the 30th USENIX Security Symposium*, 2021, pp. 2633–2650.
- [24] P. Hager *et al.*, “Evaluation and mitigation of the limitations of large language models in clinical decision-making,” *Nature Medicine*, vol. 30, no. 9, pp. 2613–2622, 2024, doi: 10.1038/s41591-024-03097-1.
- [25] B. Meskó and E. J. Topol, “The imperative for regulatory oversight of large language models (or generative AI) in healthcare,” *npj Digital Medicine*, vol. 6, no. 1, 2023, doi: 10.1038/s41746-023-00873-0.
- [26] L. Yan *et al.*, “Practical and ethical challenges of large language models in education: a systematic scoping review,” *British Journal of Educational Technology*, vol. 55, no. 1, pp. 90–112, Jan. 2024, doi: 10.1111/bjet.13370.
- [27] OpenAI, “GPT-4 system card,” *OpenAI*, pp. 41–100, 2023.
- [28] MetaAI, “Meta and Microsoft introduce the next generation of Llama,” *Meta*. 2023. [Online]. Available: <https://about.fb.com/news/2023/07/llama-2/>
- [29] J. Hoffmann *et al.*, “An empirical analysis of compute-optimal large language model training,” *Advances in Neural Information Processing Systems*, no. NeurIPS 2022, 2022.
- [30] Anthropic, “The claude 3 model family: opus, sonnet, haiku anthropic,” *Anthropic*, pp. 1–42, 2024.
- [31] S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal, “Detecting hallucinations in large language models using semantic entropy,” *Nature*, vol. 630, no. 8017, pp. 625–630, 2024, doi: 10.1038/s41586-024-07421-0.
- [32] L. Zhou, W. Schellaert, F. M.-Plumed, Y. M.-Daval, C. Ferri, and J. H.-Orallo, “Larger and more instructable language models become less reliable,” *Nature*, vol. 634, no. 8032, pp. 61–68, 2024, doi: 10.1038/s41586-024-07930-y.
- [33] M. Steyvers *et al.*, “What large language models know and what people think they know,” *Nature Machine Intelligence*, vol. 7, no. 2, pp. 221–231, 2025, doi: 10.1038/s42256-024-00976-7.
- [34] A. Vaswani *et al.*, “Attention is all you need,” in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, 2017, pp. 1–11.
- [35] Y. Xiong *et al.*, “Nyströmformer: a Nyström-based algorithm for approximating self-attention,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, pp. 14138–14148, 2021, doi: 10.1609/aaai.v35i16.17664.
- [36] M. Bakyt, L. A. Spada, K. Moldamurat, Z. Kadirbek, and F. Yermekov, “Review of data security methods using low-earth orbiters for high-speed encryption,” in *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems*, 2024, pp. 1366–1375, doi: 10.1109/ICUIS64676.2024.10867245.




- [37] A. Rawal, J. Rawal, A. Raglin, Q. Wang, and Z. Tang, "Causal reasoning with large language models – a ChatGPT case study," in *Artificial Intelligence in HCI (HCII 2025)*, 2025, pp. 358–378, doi: 10.1007/978-3-031-93429-2_24.
- [38] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret, "Transformers are RNNs: fast autoregressive transformers with linear attention," in *37th International Conference on Machine Learning, ICML 2020*, 2020, pp. 5112–5121.
- [39] M. Bakyt *et al.*, "Advanced cybersecurity framework for LEO aerospace: integrating quantum cryptography, artificial intelligence anomaly detection, and blockchain technology," *Journal of Robotics and Control*, vol. 6, no. 2, pp. 695–714, 2025, doi: 10.18196/jrc.v6i2.25918.
- [40] K. Moldamurat, Y. Seitkulov, S. Atanov, M. Bakyt, and B. Yergaliyeva, "Enhancing cryptographic protection, authentication, and authorization in cellular networks: a comprehensive research study," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 1, pp. 479–487, 2024, doi: 10.11591/ijece.v14i1.pp479-487.
- [41] J. Shobana and M. Murali, "Abstractive review summarization based on improved attention mechanism with pointer generator network model," *Webology*, vol. 18, no. 1, pp. 77–91, 2021, doi: 10.14704/WEB/V18I1/WEB18028.
- [42] A. Kumar, S. Seth, S. Gupta, and S. Maini, "Sentic computing for aspect-based opinion summarization using multi-head attention with feature pooled pointer generator network," *Cognitive Computation*, vol. 14, no. 1, pp. 130–148, 2022, doi: 10.1007/s12559-021-09835-8.
- [43] W. Li, R. Peng, Y. Wang, and Z. Yan, "Knowledge graph based natural language generation with adapted pointer-generator networks," *Neurocomputing*, vol. 382, pp. 174–187, 2020, doi: 10.1016/j.neucom.2019.11.079.
- [44] D. Pandey and C. R. Chowdary, "Modeling coherence by ordering paragraphs using pointer networks," *Neural Networks*, vol. 126, pp. 36–41, 2020, doi: 10.1016/j.neunet.2020.02.022.
- [45] T. Ouchi and M. Tabuse, "Effectiveness of data augmentation in pointer-generator model," *Proceedings of International Conference on Artificial Life and Robotics*, vol. 25, pp. 390–393, 2020, doi: 10.5954/ICAROB.2020.OS16-1.
- [46] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 3045–3059, doi: 10.18653/v1/2021.emnlp-main.243.
- [47] T. Ouchi and M. Tabuse, "Comparison of data augmentation methods in pointer-generator model," *Journal of Robotics, Networking and Artificial Life*, vol. 8, no. 2, 2021, doi: 10.2991/jmal.k.210713.003.
- [48] W. Wei, J. Wu, and C. Zhu, "Special issue on deep learning for natural language processing," *Computing*, vol. 102, no. 3, pp. 601–603, 2020, doi: 10.1007/s00607-019-00788-3.

BIOGRAPHIES OF AUTHORS






Aknur Kossayakova    is Ph.D. in Engineering. She is Associate Professor of the Department of Information and Communication Technologies at North Kazakhstan University named after M. Kozybayev, Kazakhstan, Petropavlovsk, Pushkin St., 86. She can be contacted at email: aknur_ast_enu@mail.ru.






Kurmashev Ildar    is Ph.D. in Engineering. He is Associate Professor of the Department of Information and Communication Technologies at North Kazakhstan University named after M. Kozybayev, Kazakhstan, Petropavlovsk, Pushkin St., 86. He can be contacted at email: ikurmashev@ku.edu.kz.






Luigi La Spada    received his Ph.D. in Electronic Engineering at 2014 from University of Pennsylvania (UPenn, Philadelphia, USA) and RomaTre University (Rome, Italy). From November 2018, he is in the School of Engineering and the Built Environment at Edinburgh Napier University as Assistant Professor in Electrical and Electronic Engineering. His research received international scientific recognition and high distinction on several media press (i.e. CNN, CBS, Times, and Aspen Institute). He can be contacted at email: L.LaSpada@napier.ac.uk.






Nida Zeeshan    is experienced in computing and cybersecurity domains while serving various HEI(s) in academic and research roles. She is a Microsoft Azure AI-900 Certified from Microsoft, and also associated with HEA (UK) as an Associate Fellowship. She has earned her M.Sc. in Computer Networks and Security from the University of Essex with Merit. Her thesis title was “PKI-based digital certificate authentication framework for the internet of things (IoT)”. She can be contacted at email: nida.zeeshan@napier.ac.uk.






Makhabbat Bakyt    received her Bachelor of Engineering and Technology and Master of Engineering from the L.N. Gumilyov ENU, Astana, Kazakhstan. She is currently a doctoral student of the Department of Information Security, IT Faculty at the L.N. Gumilyov ENU. Her interests include next research area: aircraft data encryption, cryptographic protection, information security. She can be contacted at email: bakyt.makhabbat@gmail.com.



Moldamurat Khuralay    was educated at the I. Zhansugurova Zhetysu State University, specialist physics and informatics. Academy of Economics and Law named after academician U.A. Dzholdasbekov. Bachelor of the specialty Finance, Turkish State University, Ankara, 2008, 2010 Candidate of Technical Sciences, the MSHE of the RK, at the NSA at the Institute of Mathematics at OD53.12. on the topic: verification and automation of microcontroller programming, the dissertation is scientifically defended. Currently, she is Associate Professor of the Department of Space Technique and Technology at the L.N. Gumilyov ENU, Astana, Kazakhstan. Her research interests include IT technologies, radio engineering, programming of microcontrollers and automation systems, and modern technologies for designing space nanosatellites. She can be contacted at email: moldamurat@yandex.kz.



Omirzak Abdirashev    is Ph.D., acting Associate Professor of the Department of “Space Engineering and Technology” at the Eurasian National University named after L.N. Gumilyov, Astana, Kazakhstan. His scientific interests cover spacecraft, fundamentals of hydraulics and hydropneumatic actuators for aircraft, satellite communication systems, and fundamentals of rocket and space technology. He can be contacted at email: omeke_92@mail.ru.