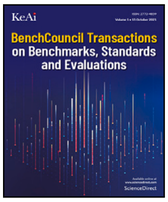
Contents lists available at ScienceDirect

BenchCouncil Transactions on Benchmarks, Standards and Evaluations

journal homepage: www.keaipublishing.com/en/journals/benchcouncil-transactions-on-benchmarks-standards-and-evaluations/



Full length article

Evaluatology's perspective on AI evaluation in critical scenarios: From tail quality to landscape

Zhengxin Yang *

University of Chinese Academy of Sciences, Beijing, China
Research Center Of Distributed Systems, State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Keywords:
Tail Quality
Evaluatology
AI inference

ABSTRACT

Tail Quality, as a metric for evaluating AI inference performance in critical scenarios, reveals the extreme behaviors of AI inference systems in real-world applications, offering significant practical value. However, its adoption has been limited due to the lack of systematic theoretical support. To address this issue, this paper analyzes AI inference system evaluation activities from the perspective of Evaluatology, bridging the gap between theory and practice. Specifically, we begin by constructing a rigorous, consistent, and comprehensive evaluation system for AI inference systems, with a focus on defining the evaluation subject and evaluation conditions. We then refine the Quality@Time-Threshold (Q@T) statistical evaluation framework by formalizing these components, thereby enhancing its theoretical rigor and applicability. By integrating the principles of Evaluatology, we extend Q@T to incorporate stakeholder considerations, ensuring its adaptability to varying time tolerance. Through refining the Q@T evaluation framework and embedding it within Evaluatology, we provide a robust theoretical foundation that enhances the accuracy and reliability of AI system evaluations, making the approach both scientifically rigorous and practically reliable. Experimental results further validate the effectiveness of this refined framework, confirming its scientific rigor and practical applicability. The theoretical analysis presented in this paper provides valuable guidance for researchers aiming to apply Evaluatology in practice.

1. Introduction

With the rapid advancement of artificial intelligence (AI) technologies, evaluating AI inference systems has become increasingly critical. These systems operate in dynamic and unpredictable environments, ranging from the online deployment of large-scale language models such as ChatGPT [1,2] to real-time applications in autonomous driving [3–5] and smart healthcare [6,7]. The increasing reliance on AI in these critical domains introduces significant challenges. For instance, online real-time recommendation systems are crucial in e-commerce and content streaming platforms, directly affecting user engagement and satisfaction. Delays in inference can lead to user impatience and churn, impacting the overall effectiveness of these systems. Similarly, autonomous driving, a safety-critical domain, requires real-time decision-making for tasks such as object detection and lane detection [8], with even minor errors posing catastrophic risks to safety [9,10]. These challenges necessitate the development of reliable and objective methods to evaluate the inference performance of AI systems.

However, traditional evaluation methods often face two major issues. First, evaluations frequently rely on isolated metrics, such as accuracy or inference throughput, without accounting for the complex interactions between various factors [11]. These single-dimensional evaluations fail to capture the true performance of AI inference systems, especially when real-world, dynamic conditions are taken into account. Second, AI inference system evaluation remains a complex and uncertain process due to the inherent intricacies of computer systems [12] and the absence of well-established, interpretable theories—particularly for systems equipped with neural networks [13,14]. Without clear, theoretically grounded evaluation criteria, existing methods often fall short, relying on industry standards without comprehensive theoretical analysis [15–18]. This theoretical gap poses challenges for researchers trying to evaluate AI inference systems, often leading to questions about the reliability of existing evaluation methods.

To address the issues, **Quality@Time-Threshold** (Q@T) was introduced by Yang et al. [11] as a metric designed to measure how an AI inference system's quality fluctuates under strict time constraints. By

* Correspondence to: Research Center Of Distributed Systems, State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China.

E-mail address: yangzhengxin17z@ict.ac.cn.

https://doi.org/10.1016/j.tbench.2025.100203
Received 24 January 2025; Received in revised form 6 April 2025; Accepted 28 April 2025
Available online 27 May 2025

considering both inference time and quality, Q@T provides a statistical evaluation framework that captures the "**Tail Quality**" phenomenon—extreme fluctuations in inference quality that are often overlooked by traditional methods [15,18]. This is especially crucial in critical applications such as autonomous driving and medical diagnostics, where poor performance could lead to severe consequences [11]. While Q@T offers an important step forward in AI evaluation, it still faces challenges due to the absence of a solid theoretical foundation, particularly the lack of guidance from Evaluatology. This theoretical gap raises concerns about the reliability and rigor of Q@T, especially when applied in diverse scenarios. Furthermore, Q@T's applicability is limited in contexts where flexible time constraints are necessary, as it is more suited for strict time thresholds. Thus, the need for a more robust theoretical framework becomes evident, one that can guide the evaluation process in a scientifically grounded way.

In this paper, we aim to address these challenges by analyzing Q@T from the perspective of **Evaluatology**, a formal science of evaluation. Evaluatology offers a systematic methodology for modeling and understanding complex systems, providing a theoretical foundation for improving evaluation methods. By applying Evaluatology's five core axioms and standard evaluation methodology, we redefine and clarify the core components of the evaluation system in Q@T—namely, the evaluation subject and evaluation conditions. This approach helps overcome the limitations of Q@T, ensuring that it is both scientifically rigorous and adaptable to different application scenarios, such as autonomous driving, healthcare, and e-commerce.

This paper's main contributions are as follows:

- **Theoretical Validation of Q@T:** We integrate Evaluatology's principles with Q@T to provide a stronger theoretical foundation for evaluating AI inference systems.
- **Bridging Theory and Practice:** We bridge the gap between theoretical foundations (Evaluatology) and the practical evaluation of AI systems (Q@T).
- **Refining Q@T:** We refine the Q@T framework by introducing a more flexible approach that incorporates stakeholder needs and time constraints, enhancing its versatility across real-world applications.

2. Background

This section introduces two key aspects: Evaluatology [19–21] and Q@T [11]. Section 2.1 introduces the five core axioms of Evaluatology and the standard evaluation methodology. Section 2.2 briefly introduces the definition of the Q@T evaluation metric and the statistical evaluation framework for assessing Q@T in AI inference systems.

2.1. Evaluatology

This section introduces the five core axioms of Evaluatology and the standard evaluation methodology. These are the two most fundamental components of Evaluatology, providing a rigorous theoretical foundation for evaluating complex systems such as AI inference systems. Applying these principles, we can reanalyze Q@T and ensure its evaluation framework is scientifically sound and practically applicable.

2.1.1. The five core axioms of Evaluatology

The five core axioms of Evaluatology form the foundation of its evaluation methodology, detailed as follows:

- **The Axiom of the Essence of Composite Evaluation Metrics**: A composite evaluation metric either has inherent physical meaning or is defined by a value function that combines base quantities.
- **The Axiom of True Evaluation Outcomes**: For a well-defined evaluation system, when well-defined evaluation conditions are applied to a well-defined evaluation subject, its evaluation outcomes must possess true values.
- **The Axiom of Evaluation Traceability**: For the same evaluation subject, differences in evaluation outcomes can be attributed to variations in the evaluation conditions (ECs), ensuring traceability and interpretability of evaluation outcomes.
- **The Axiom of Comparable Evaluation Outcomes** Evaluation outcomes are comparable only evaluation subjects are evaluated under equivalent evaluation conditions (EECs).
- **The Axiom of Consistent Evaluation Outcomes**: Evaluation outcomes from different samples within a population of evaluation conditions consistently converge toward the true evaluation outcomes of the entire population.

By adhering to these axioms, the evaluation methodology provides a rigorous, reliable, and comparable framework for evaluating complex systems, including AI inference systems, ensuring consistency, accuracy, and reliability across various application scenarios.

2.1.2. The standard evaluation methodology

In Evaluatology, the standardized evaluation methodology consists of four key steps. These are summarized as follows:

Defining and characterizing the subject:. The first step is clearly defining and describing the subject to be evaluated. A well-defined subject is essential for valid comparisons between different instances of the same subject definition. This stage also involves modeling the subject, which includes outlining its detailed structure. A rigorous definition and consensus on the subject's model among stakeholders are crucial for ensuring the validity of the evaluation process.

Defining and clarifying the evaluation system (ES):. The second step is constructing a minimal yet complete Evaluation System (ES) that operates autonomously. This is a crucial component of the evaluation process and must meet two main criteria: it must function independently and encompass all the essential factors needed for the evaluation task. It is important to note that any changes to the independent factors in the ES will impact the final evaluation results. Defining the ES comprehensively is challenging because too many factors can lead to excessive evaluation costs, while too few may fail to capture all the critical influences on the results.

Acquiring the evaluation conditions (ECs):. Once the ES is defined, the next step is establishing Evaluation Conditions (ECs). These conditions are derived by isolating the subject from the ES. Defining these conditions ensures the evaluation process is based on realistic and controlled parameters.

Determining the evaluation methodologies:. After defining the ES and ECs, the final step is to analyze the nature of the ES and determine the appropriate evaluation methodologies. The key goal in this phase is to ensure that the evaluation methods meet the standards of Evaluatology's five core axioms, ensuring the comprehensiveness and reliability of the evaluation process.

In conclusion, the standardized evaluation methodology in Evaluatology involves clearly defining the evaluation subject, establishing the evaluation system, defining the evaluation conditions, and selecting the appropriate evaluation methods. These steps ensure that complex systems, such as AI inference systems, are evaluated rigorously and systematically. The evaluation process can produce consistent, accurate, and comparable results by following these steps.

2.2. Q@T and tail quality

The Quality@Time-Threshold (Q@T) metric was proposed to measure the ability of an AI inference system to maintain stable high inference quality under strict time constraints. This is of practical significance, as an AI inference system with high-quality predictions should achieve high and stable inference quality even under lower time thresholds.

2.2.1. Definition of Q@T

Q@T evaluates AI inference systems by balancing inference quality and time constraints. Given a dataset $D = \{x_i, y_i\}_{i=1}^n$, where x_i represents the input and y_i represents the ground truth, the model M generates prediction $y_i' = M(x_i)$. The overall inference quality q is determined by comparing $Y' = \{y_i'\}_{i=1}^n$ with the ground truth $Y = \{y_i\}_{i=1}^n$ using a quality evaluation function (e.g., accuracy or F-score). To account for the impact of inference time, the validity of each inference result is determined by whether the inference time exceeds a specified time threshold θ. This leads to the following equation for Q@T:

$$q_\theta = \text{evaluate}(\{M(x_i) \cdot \mathbf{1}_\theta + \texttt{error} \cdot (1 - \mathbf{1}_\theta)\}_{i=1}^n, \{y_i\}_{i=1}^n), \tag{1}$$

where $\mathbf{1}_\theta$ is an indicator function that returns 1 if the inference of x_i completes within the threshold θ, and 0 otherwise. The placeholder `error` denotes a default output substituted when the inference time exceeds the threshold, thereby effectively invalidating the corresponding model output. The function $\text{evaluate}(\cdot, \cdot)$ computes a quality metric — such as accuracy — between the (potentially `error`-substituted) model outputs and the corresponding ground-truth labels.

Q@T offers a comprehensive evaluation that is especially useful for real-time applications where quality and time are critical.

2.2.2. Statistical evaluation framework for Q@T

In AI inference systems, inference time can vary significantly due to factors such as hardware configurations, deep learning frameworks, and data processing pipelines. This variability impacts the estimation of Q@T, as fluctuations in inference time influence quality under specific time constraints.

To evaluate Q@T accurately, the statistical framework models inference time as a random variable T, which follows an unknown distribution $\mathcal{D}$ and is influenced by various system components. The Q@T metric becomes a random variable Q dependent on T and the system components C^i, expressed as a conditional probability distribution:

$$Q_\theta = f(T \mid \theta, C^1, C^2, \dots), \quad T \sim \mathcal{D}. \tag{2}$$

The framework uses the Monte Carlo simulation to collect inference time samples and Kernel Density Estimation (KDE) to estimate the distribution of these times. This non-parametric approach avoids assumptions about the form of the distribution. Convergence is monitored using Jensen–Shannon Divergence (JSD), stopping the simulation when the distribution stabilizes.

The steps include:

- **Sampling Inference Time:** Collect inference time samples using Monte Carlo simulations across multiple rounds.
- **Kernel Density Estimation (KDE):** Apply KDE to estimate the probability density function $\hat{f}(t)$ of inference time.
- **Convergence Check using Jensen–Shannon Divergence (JSD):** Calculate the JSD between distributions from different sample sizes, stopping when the JSD is sufficiently small, indicating convergence.

After convergence is achieved, the framework performs N independent evaluation trials to quantify the quality metric under the stabilized distribution. Each trial produces a sample q_i from the random variable Q, resulting in a set of observations $\{q_i\}_{i=1}^N$.

The final Q@T metric is computed as a statistical characterization of Q_θ when the time threshold is set to $\theta = \text{T}$, based on these samples:

$$\text{Q@T} = \mathcal{S}_{\theta=\text{T}}(\{q_i\}_{i=1}^N), \quad q_i \sim Q_{\theta=\text{T}}, \tag{3}$$

where $\mathcal{S}_{\theta=\text{T}}(\cdot)$ denotes a set of statistical characteristics such as the sample mean, variance, and quantiles. This formulation enables a comprehensive representation of inference quality under time constraints, going beyond reliance on a single observation or the expected value alone.

Once a reliable distribution is obtained, the framework computes the final Q@T values, reflecting both the variability in inference time and the extreme fluctuations in inference quality.

2.2.3. The extreme tail quality phenomenon

The Tail Quality phenomenon arises from the statistical evaluation framework in Q@T, which generates a distribution of inference quality values instead of a single-point estimate. Tail Quality specifically refers to the extremely low-quality values observed at the tail of this distribution. This highlights how Q@T can reveal performance variations that traditional evaluation methods might overlook.

This phenomenon underscores the importance of Q@T in evaluating AI systems, especially in critical or real-time applications, where such extreme deviations could have serious consequences. This makes Q@T a valuable tool for assessing AI systems, it provides a more comprehensive understanding of the system's inference ability under strict time constraints.

However, a limitation of Q@T is that it is focused on evaluating systems within a predefined time threshold. In cases where time constraints are less stringent, Q@T may not be as applicable. This limitation will be addressed in Section 4, where potential extensions and improvements to the Q@T framework are discussed, making it more versatile and suitable for a wider range of scenarios.

3. Reanalyzing AI inference evaluation from the perspective of evaluatology

In the context of Evaluatology, an evaluation system (ES) is defined as the smallest autonomous, self-contained system capable of operating automatically. The evaluation system can be broken down into two parts: evaluation subject and evaluation conditions. The evaluation subject refers to the "thing" being evaluated [20], which can be either an individual or a system. This is the core of the entire evaluation framework [19]. By precisely defining the subject, we can distinguish which part of the evaluation system the final evaluation outcomes belong to. The evaluation conditions, on the other hand, encompass all factors of the evaluation system other than the subject, serving as the primary determinants that influence the evaluation outcomes. When evaluating AI inference systems, defining both the evaluation subject and conditions presents challenges due to the inherent complexity of AI and computer systems. Therefore, in this section, we will follow a process where we first clearly define the evaluation system (Section 3.1), then separate the evaluation subject from the system (Section 3.2), and finally, establish the evaluation conditions (Section 3.3).

3.1. Clarifying primary components constitute evaluation system

This section provides a detailed discussion and analysis of the components that make up the evaluation system (ES) in the context of evaluating AI inference systems. The primary purpose of AI inference systems is to make predictions across various AI inference activities. Therefore, the key to constructing the evaluation system (ES) is to understand the structure of these inference activities. AI inference activities are complex and multifaceted, involving various components that interact with one another. These activities are hierarchical and require several components to work in concert to produce accurate predictions. This inherent complexity is a central challenge in evaluating AI inference systems. Below, we provide a concrete definition of the evaluation system, the primary framework we use to define the evaluation systems for AI inference tasks. Fig. 1 also illustrates the general structure of the entire framework.

Application Scenarios & Tasks: First, we must define the highest level of the evaluation system, which involves clarifying the problem definition for the AI inference activities. This includes identifying the application scenario and the tasks the AI inference activities are expected to perform. The definition of the application scenario and tasks is crucial in determining the evaluation criteria and metrics for the entire evaluation. For example, in e-commerce applications, the most important task for the AI inference system is often recommendation, where recommendation accuracy and latency are the core evaluation

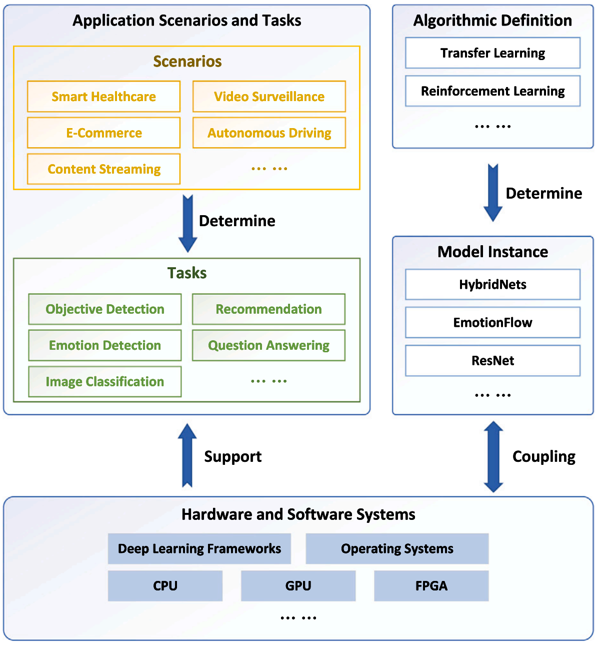


Fig. 1. An overview of the main components that make up the evaluation system in the context of evaluation AI inference systems. The diagram illustrates a hierarchical relationship, progressing from application scenarios and tasks (top left) to algorithmic definitions (top right) and then to specific model instances (middle right) and their execution environments (bottom). Each layer influences the instantiation of the next: application requirements determine which tasks are relevant, which in turn guide the selection of algorithmic approaches and models. The arrow labeled “Determine” indicates downward decisions or constraints, while the arrow labeled “Support” denotes foundational infrastructure (e.g., hardware/software platforms supporting model execution). “Coupling” highlights tight interdependencies between components, such as how models and algorithms must adapt to the capabilities and constraints of the hardware/software stack. This structure underscores the complexity of defining the evaluation subject, where inference performance emerges from interactions across all layers.

metrics [22]. However, in autonomous driving, the evaluation system must account for real-time tasks such as object detection, drivable area segmentation, and lane line detection [8]. Given the critical nature of real-time decision-making in this scenario, time constraints become particularly important, and the evaluation system must emphasize the inference time.

Algorithmic Definition: After identifying the AI inference system’s application scenario and tasks, the next step is to clarify the algorithmic definition required to accomplish these tasks. This is a critical layer in the evaluation system as the algorithms form the foundation for the system’s inference capabilities. Since this evaluation is focused on assessing the inference level of AI systems, we concentrate on neural network-based algorithms. Additionally, these algorithms must be clearly defined, including the input and output data requirements, which are typically influenced by both the application scenario and the specific tasks.

For example, in the driver assistance scenario, the system needs to identify abnormalities in the driver’s facial expressions and voice to prevent potential accidents [23,24]. In the smart healthcare scenario, emotion recognition might focus on monitoring the patient’s bioelectrical signals to avoid unforeseen incidents during medical procedures [23,25]. In both cases, while the emotion recognition remains the same, the specific data sources and algorithmic needs vary due to the differing application contexts.

Model Instances: Next, the core implementation of the algorithm must be defined. This involves specifying which neural network model will implement the task objectives. The choice of neural network model directly impacts how the algorithm processes input and output data. The chosen model must also account for environmental factors and data variances in real-world applications. For instance, if the system encounters rain or snow in autonomous driving, the model trained on a general dataset might require additional complex data processing for the images captured by the vehicle’s cameras. However, suppose the model is trained using a specialized dataset that includes weather-related variations [5,26,27]. It may not require these additional processing steps and could still achieve accurate object detection in adverse weather conditions.

Hardware & Software Systems: Finally, to support the algorithm’s functioning, complete hardware and software infrastructure are necessary to run the neural network model efficiently. The hardware configuration typically includes CPUs, GPUs, and potentially specialized hardware such as FPGA or TPU. In real-time performance evaluations, hardware is critical because it directly affects the inference speed. For instance, an AI model might perform well on a powerful GPU but may face delays on a CPU, especially in time-sensitive applications [28]. Furthermore, hardware selection must also consider other factors such as energy consumption, size, and form factor, particularly for embedded systems. On the software side, this includes libraries, frameworks, and operating systems that facilitate the execution of AI models, such as TensorFlow [29], PyTorch [30], and others. These frameworks optimize the computation processes of the neural network model but interact directly with the hardware, and variations in the specific configurations and versions used can result in differences in accuracy and performance [31].

In summary, we have analyzed several primary factors the evaluation system contains: the application scenario, evaluation metrics, data processing, algorithm selection, and computer hardware/software infrastructure. These factors are interdependent and form a complete evaluation system for AI inference activities [15,32]. However, a key challenge remains in clearly delineating the boundaries between the evaluation subject and the evaluation conditions. This is a critical issue for understanding how different components of the evaluation system interact and how their influence on the final evaluation outcome can be isolated and measured.

3.2. *Defining and identifying evaluation subject*

Clarifying the evaluation subject is essential for a rigorous evaluation, as it directly determines the boundaries within which the evaluation outcomes will be interpreted. However, this task is not trivial in AI inference systems evaluation. This is particularly evident in the work of Yang et al. [11], where the definition of the AI inference system itself is somewhat ambiguous. This lack of a clear and universally agreed-upon definition poses challenges in accurately assessing the performance of AI inference systems in complex environments.

The reason for this ambiguity stems from the inherent complexity of the evaluation system [12,32]. As shown in Fig. 1, the evaluation system comprises multiple interrelated components, each of which affects the system’s overall performance. A neural network model can be seen as an instantiation of an algorithm designed to solve a particular task and as a component that must be integrated seamlessly with the computational hardware and software environment. The interaction between these components creates a complex, tightly coupled system. Therefore, a central question in evaluatology for AI inference systems is whether or not the neural network model should be considered part of the evaluation subject.

3.2.1. *Definition of evaluation subject: Excluding model instance*

One potential approach to defining the evaluation subject is to exclude the neural network model instance and treat only the computational hardware and software system as the subject being evaluated. As illustrated in Fig. 2, the model instance is considered an input to the evaluation subject, which processes it within a defined task.

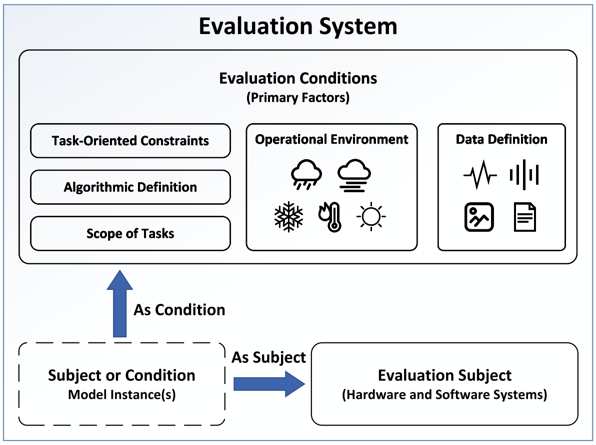


Fig. 2. The composition of the Evaluation System, including the Primary Factors as Evaluation Conditions, the Hardware and Software system as the Evaluation Subject, and the Model Instance, which may be considered part of the Subject or the Condition, depending on the specific evaluation objectives.

From this perspective, the evaluation is directed at understanding the general capability of the AI inference system to perform inference tasks rather than measuring the specific performance of the neural network model and a computer system. This means that different implementations of the same task — such as two distinct models solving the same problem — would not affect the evaluation, as the focus is on the hardware–software system and its ability to process data efficiently. For example, in autonomous driving, the evaluation would focus on the system's ability to process input data from sensors and generate timely decisions independent of the specific model implementation used for object detection.

3.2.2. Definition of evaluation subject: Including model instance

The second definition of the evaluation subject involves treating both the neural network model and the hardware/software system as part of the evaluation subject, as shown in Fig. 2. Here, the specific model implementation (e.g., a particular deep learning network) becomes a key factor in determining the system's ability to execute inference tasks. In this case, the evaluation would measure the system's overall inference capability, including inference quality and inference time, while considering the effects of different model implementations. This does not mean that the first definition cannot measure the system's overall performance, but rather that eliminating the impact of model instances would require greater costs to measure the overall performance that is universally applicable under specific tasks. For example, a model trained with a specialized dataset for handling weather-related changes in autonomous driving would perform differently than a model trained on a generic dataset.

3.3. Establishing evaluation conditions

According to Evaluatology, evaluation conditions (ECs) are determined by isolating the evaluation subject within the evaluation system. ECs encompass all external factors that influence the performance of the AI inference system. Therefore, in this study, the evaluation conditions (ECs) naturally include practical application scenarios, tasks, algorithmic definitions, and the model itself, depending on whether the model instance is considered part of the evaluation subject. Specifically, based on the analysis in Section 3.1, these components define the primary factors that must be considered within the EC, as detailed in the following sections.

Operational Environment: In autonomous driving applications, factors such as weather (rain, snow, fog), traffic density, and road conditions can significantly affect the evaluation outcomes. The system must process complex sensor data in real time while also meeting the computational demands of the model's inference, partly due to the intrinsic complexity of the subject itself. To address this, by isolating the subject and removing confounding factors through statistical methods within the ECs, we can focus on understanding how specific environmental factors impact the evaluation results. As a result, ECs must account for these environmental variations and simulate different conditions to assess the system's robustness.

Data Definition: In real-world applications, AI systems must handle data that can be multimodal, noisy, or incomplete. For example, in autonomous driving, the system may receive data from multiple sensors, such as cameras and LiDAR systems. Thus, the ECs must clearly define the form of input data in the evaluation process, ensuring that data quality is consistently considered. How data from each sensor is processed, combined, and interpreted is crucial for accurately assessing the system's overall outcomes.

Task-Oriented Constraints: Tasks like object detection in autonomous driving have strict real-time performance requirements, whereas tasks like clinical diagnostics may not have stringent time constraints but must still meet high-precision standards. In the context of Q@T, the time threshold θ is a central evaluation condition used to determine whether the system's inference time is acceptable. If the inference time exceeds this threshold, the system's performance may still be considered inadequate, even if the inference quality is high. Therefore, ECs must include these real-time constraints, which may vary depending on the application's requirements.

Scope of Tasks: Additionally, we propose that the scope of tasks for evaluation should be appropriately limited. While some researchers may wish to assess the general inference capability of an AI system across all possible tasks, doing so would result in an explosion of the EC space, significantly increasing evaluation costs. Moreover, we argue that evaluating an AI inference system's general processing ability across a broad range of tasks does not align with real-world application needs. Instead, focusing on specific, well-defined tasks relevant to the system's intended deployment is more practical and efficient.

Algorithmic Definition: ECs should specify the AI inference algorithm used, including whether it is based on transfer learning, reinforcement learning, or other approaches. The choice of algorithm significantly influences how input data is processed and how inference results are generated. Therefore, it is essential to include this information in the ECs to ensure consistency and comparability when evaluating the system's performance under different algorithmic conditions.

Model Instances: Lastly, if the model instance is not included in the evaluation subject, the ECs must be designed to eliminate the effects of variations in model instances on evaluation outcomes. This requires carefully controlling model configurations and ensuring that the evaluation reflects the underlying AI inference system's performance rather than the idiosyncrasies of a specific model instance.

In conclusion, by clearly defining the evaluation conditions (ECs), we ensure that the AI inference system is evaluated within comparable contexts, guaranteeing the consistency and reliability of the evaluation outcomes. This structured approach allows for a more accurate, reproducible, and objective evaluation of AI systems in complex, real-world applications, ensuring that the evaluation is scientifically rigorous and practically applicable.

4. Refining Q@T evaluation framework

In this section, we refine the Q@T evaluation framework by formalizing the evaluation subject and conditions based on the earlier analysis using Evaluatology 4.1. This formalization ensures clear definitions of the evaluation subject and conditions, providing a more rigorous and consistent framework for evaluating AI inference systems. Through experiments, we further emphasize the importance of tail quality as an indicator of extreme performance in AI inference system evaluation 4.2.

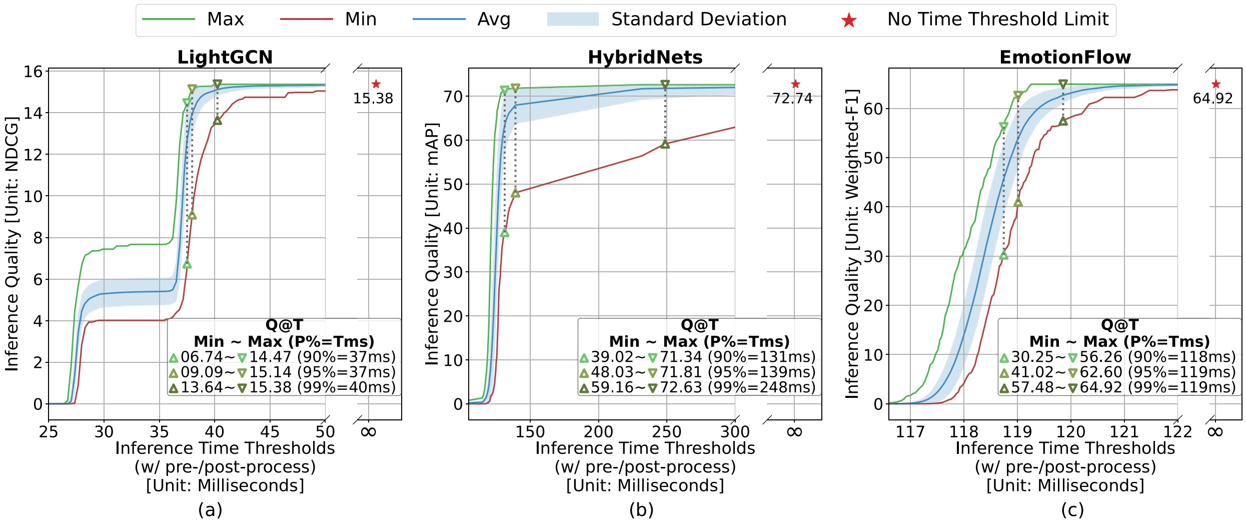


Fig. 3. Comparative Q@T evaluation under multiple inference time thresholds for three representative AI models: LightGCN (e-commerce recommendation), HybridNets (autonomous driving), and EmotionFlow (emotion recognition). The horizontal axis indicates inference latency (including data pre- and post-processing), while the vertical axis reports task-specific quality metrics (NDCG, mAP, and weighted-F1, respectively). For each model, Q@T is computed at different percentiles of inference latency, with key tail-latency points (e.g., 90%, 95%, 99%) annotated. Max/min/average Q@T scores and standard deviation are also indicated, revealing the relative performance and stability of each system under varying time constraints.

Finally, we discuss the landscape of stakeholder considerations, highlighting how incorporating the time tolerance of different stakeholders into the Q@T framework enhances the adaptability of the evaluation process 4.3.

4.1. Formalization of evaluation subject and conditions

From the perspective of the Q@T framework, Yang et al. [11] adopts the second definition of the evaluation subject, as described in Section 3.2, where both the model instance and the computational infrastructure (hardware/software system) are considered a unified entity. This approach aligns with Q@T's focus on evaluating the overall AI inference system, especially where the model and hardware are tightly integrated. However, the Q@T framework is flexible enough to accommodate the first definition of the evaluation subject, where the model instance is excluded. By sufficiently sampling the model instance space, Q@T can eliminate the impact of specific model instances on the evaluation outcomes, inspired by Evaluatology. This adaptability allows Q@T to evaluate AI systems from different perspectives, depending on the goals of the evaluation.

Building on this flexibility, the Q@T evaluation framework establishes a causal relationship between the system configuration C^i, inference time T, and quality Q, as shown in Eq. (2). This relationship allows for the assessment of the AI inference system's performance across varying configurations, even though Yang et al. [11] does not explicitly define the evaluation subject.

Formalization Given the inherent complexity of the evaluation subject, the evaluation system must operate as a whole to objectively assess the inference capability of AI systems. This requires running the system multiple times to isolate the effects of this complexity. Statistical indicators, such as averages and confidence intervals, are essential to represent the final evaluation outcomes, mitigating the influence of confounding factors and ensuring the consistency and reliability of the evaluation process. The Q@T evaluation framework addresses this challenge by incorporating statistical methods that improve the rigor of the evaluation.

While the Q@T metric was originally designed to explore the relationship between quality Q and time T, quantifying the trade-off between inference quality and time, the evaluation framework in Yang et al. [11] lacks clear definitions of the evaluation subject and conditions. Therefore, we propose distinguishing the elements within C^i into **primary factors** of evaluation conditions and **inherent factors** of the subject. Specifically, primary factors should be considered as factors $\mathcal{P}^i$ that influence the evaluation outcomes **and can be actively identified and controlled. In contrast**, elements belonging to the subject should be regarded as inherent factors $\mathcal{I}^j$, which may affect the evaluation results due to system complexity but are difficult to observe or isolate explicitly. Therefore, the original evaluation framework, as modeled by Eq. (2) can be reformulated as the following equation:

$$Q_\theta = f(T \mid \theta, \{\mathcal{P}^i\}_{i=1}^n; \{\mathcal{I}^j\}_{j=1}^m), \quad T \sim \mathcal{D}. \tag{4}$$

This distinction reflects a fundamental challenge in evaluation design: Although the identification and control of primary factors enable the construction of a self-contained evaluation system, the presence of inherent factors introduces variability that cannot be eliminated through experimental control alone. Instead, their influence must be mitigated through statistical methods. The statistical evaluation framework for Q@T thus plays a critical role in ensuring that the evaluation results remain robust and generalizable, despite the uncontrollable complexity of the inference system.

Based on the revised formulation in Eq. (4), we can derive practical principles for carrying out the evaluation process. When analyzing these factors, it is essential to vary one condition at a time — either from $\mathcal{P}^i$ or $\mathcal{I}^i$ — to isolate the effects of each. In practice, however, because inherent factors $\mathcal{I}^i$ are often unobservable or difficult to control, statistical methods must be applied first to reduce their influence. This requires keeping the ECs $\mathcal{P}^i$ fixed during repeated measurements, enabling the identification and mitigation of inherent variability. Only after this stabilization can we vary the ECs to meaningfully compare evaluation outcomes across different settings. Following this principle ensures that the evaluation process remains consistent and comparable, adhering to the five axioms of Evaluatology and guaranteeing the reliability and objectivity of the evaluation results.

Experiments We conducted three groups of experiments across diverse AI tasks to evaluate the effectiveness and generalizability of our proposed evaluation framework. As shown in Fig. 3, these experiments illustrate how Q@T varies under different inference time thresholds:

- **E-commerce Recommendation (LightGCN** [22]**)**: As shown in Fig. 3(a), Q@T effectively tracks the changes in NDCG (Normalized Discounted Cumulative Gain), which measures the ranking quality of

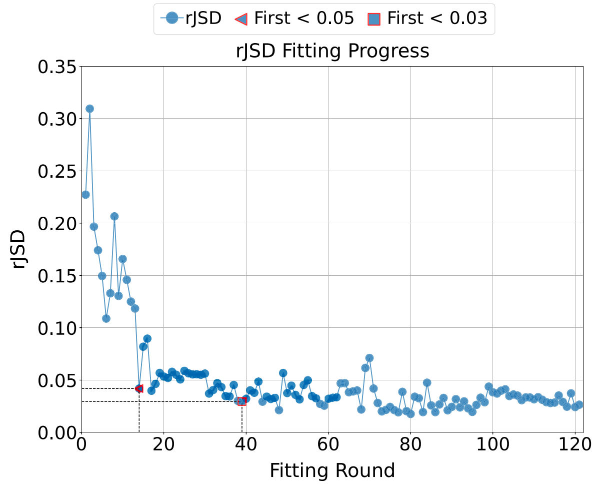


Fig. 4. Convergence validation of the revised Q@T evaluation framework, conducted on the HybridNets model for autonomous driving tasks, running on an A100 GPU. The curve shows the square root of Jensen–Shannon divergence (rJSD) over successive sampling rounds, which measures the stability of Q@T estimation as sampling progresses. The first occurrences of rJSD falling below 0.05 and 0.03 are annotated, indicating fast convergence, with rJSD dropping below the two thresholds at around the 20th and 40th rounds, respectively. This confirms the statistical robustness of the evaluation under the revised formulation.

recommendation results—the higher the value, the better the performance. These results demonstrate the applicability of our framework in time-sensitive recommendation scenarios.

- **Autonomous Driving (HybridNets** [8]): As shown in Fig. 3(b), the model performs lane line detection, traffic object detection, and drivable area segmentation. For clarity, we focus on the mAP (mean Average Precision), a widely used metric for object detection that reflects precision across categories—higher values indicate better detection accuracy. Q@T reveals how performance degrades as inference time constraints become stricter, highlighting potential quality loss in safety-critical scenarios.

- **Emotion Recognition (EmotionFlow** [33]): As shown in Fig. 3(c), Q@T captures changes in weighted-F1 score, a harmonic mean of precision and recall, under different latency thresholds—higher values represent better balance between accuracy and completeness. The experiment shows that emotion detection models can suffer from quality drops in low-latency settings.

These cross-domain experiments demonstrate the generalizability and practical utility of the revised Q@T framework under the guidance of Evaluatology, particularly in capturing time-sensitive performance variations across different AI applications.

In addition, we further verified the statistical stability of the proposed evaluation method using HybridNets in the autonomous driving domain. As shown in Fig. 4, the square root of Jensen–Shannon divergence (rJSD) gradually decreases and stabilizes as the number of sampling rounds increases. The convergence threshold of 0.05 was reached in fewer than 20 rounds, indicating that reliable statistical evaluation can still be efficiently achieved under the revised definition.

Finally, we conducted a focused analysis of EmotionFlow to explore Q@T's ability to capture tail quality. As shown in Fig. 5, we set the inference time threshold to 90% tail latency (118.75 ms) and observed the corresponding quality fluctuation. The results show that Q@T effectively reflects system performance at critical latency thresholds, offering a more comprehensive perspective for optimizing the quality-latency trade-off in AI system deployment.

Furthermore, we observed that all three models exhibited significant quality fluctuations under strict inference time constraints. This indicates that current AI software and hardware systems may still lack sufficient stability in critical scenarios, emphasizing the need for further optimization to ensure robust inference performance in real-world applications.

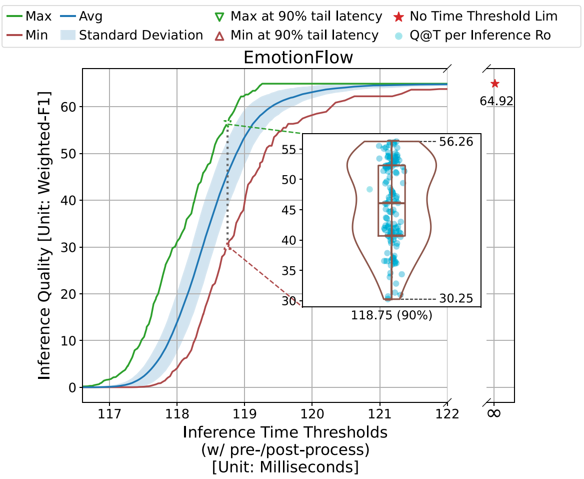


Fig. 5. Q@T evaluation results across multiple inference time thresholds using the revised Q@T framework, with the subject composed of the A100 GPU and the EmotionFlow model for emotion recognition. The violin plot represents the distribution of Q@T (measured by weighted-F1) at a specific time threshold. The box highlights the distribution when the time threshold is set to 90% tail latency (118.75 ms). Key statistics — including maximum, minimum, average, and standard deviation — are annotated, with the baseline "No Time Threshold Limit" included for reference.

4.2. The importance of tail quality

The mean value alone is insufficient to accurately capture the overall performance of AI inference systems, particularly in scenarios where extreme behavior is more indicative of the system's true performance. In real-world applications, the performance of AI systems often deviates significantly from the average in critical applications. This is the tail quality phenomenon, which Q@T specifically aims to address.

For example, in the MLPerf inference benchmark [15], the evaluation primarily focused on average- or best-quality metrics. However, this approach overlooks the performance in extreme cases where the system might fail or perform poorly, often the most critical aspect for real-world applications. Incorporating statistical methods, such as the Monte Carlo Simulation, into evaluating Q@T ensures that these extreme cases are not overlooked, providing a more comprehensive understanding of the system's true capability.

Through our experiments, we further emphasize the importance of Tail Quality in AI inference system evaluation. As shown in Fig. 5, the refined Q@T effectively captures the full spectrum of evaluation quality fluctuations, including extreme performance variations at the tail end of the distribution. For instance, when the inference time threshold is set to 118.75 ms, the lowest observed inference quality is 30.25 weighted-F1, significantly lower than the value of 64.92 obtained using traditional isolated quality metrics. The difference between these two values is approximately 2.15 times, highlighting how Q@T can reveal extreme quality fluctuations that traditional evaluation methods cannot capture. This capability ensures that the evaluation reflects not only the system's overall quality but also how it behaves under the most demanding conditions, which is critical for ensuring reliability in high-risk or real-time applications.

4.3. The landscape of stakeholder consideration

In the original Q@T evaluation framework, the focus has largely been on strict time thresholds, where inference time is a critical factor in determining system performance. However, real-world applications often involve varying levels of tolerance for inference delay, depending on the specific needs of different stakeholders. These stakeholder needs can significantly influence how the system is evaluated and what performance metrics are prioritized.

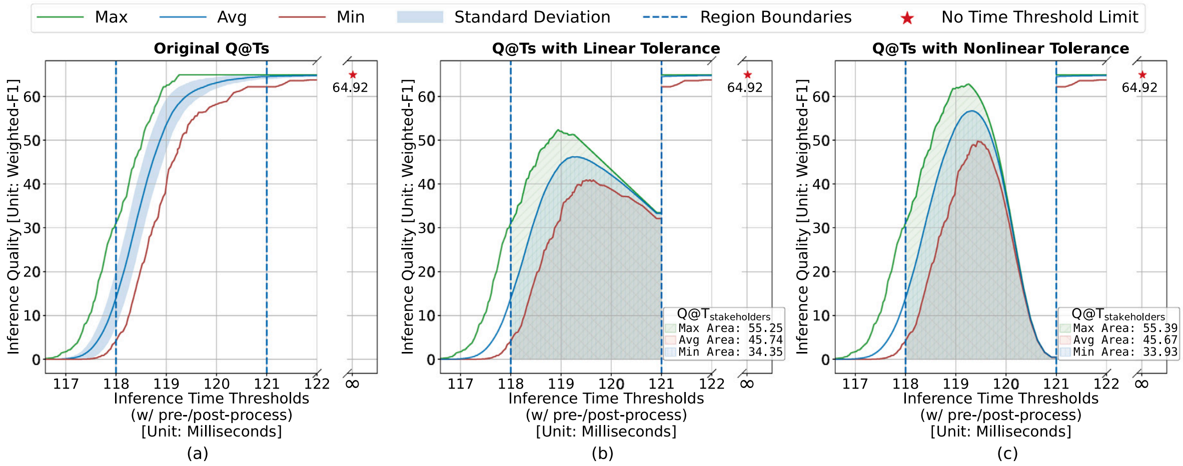


Fig. 6. Stakeholder-aware Q@T evaluation on the EmotionFlow model (weighted-F1). (a) shows the original Q@T scores under different inference time thresholds; as the threshold increases, the model is allowed more time, leading to higher Q@T values. (b) illustrates Q@T variation after applying a linear tolerance function over the tolerance interval from 118 ms to 121 ms. (c) shows the corresponding result with a nonlinear tolerance function, where high-latency points are penalized more severely.

To better address these diverse needs, we propose expanding the Q@T framework by incorporating a tolerance function that adjusts the weight of the Q@T metric based on the acceptable time thresholds for different stakeholders. This allows the evaluation framework to reflect time tolerance based on the level of urgency or flexibility that stakeholders require. The new approach ensures that the evaluation process is more adaptable to real-world constraints, where not every scenario demands the same level of urgency.

For instance, a real-time autonomous driving application may impose strict time constraints on inference, whereas in e-commerce, a slight delay beyond the threshold might be tolerable, though it could lead to user churn. However, user churn is not necessarily directly correlated with a strict time threshold, as exceeding the threshold does not automatically result in user loss. Different users have varying levels of tolerance for delay, and the relationship between user patience and inference time is not as binary or stepwise as the rigid time thresholds might suggest. In this context, the tolerance function for time delay in e-commerce applications must reflect the more gradual and nonlinear impact of delay, where mild delays may still be acceptable but could lead to different levels of consequences depending on the user's tolerance.

Building upon the original formulation of Q@T in Eq. (4), we propose a generalized version that incorporates stakeholder-specific tolerance to inference delay, referred to as Stakeholder-aware Q@T:

$$\mathrm{Q@T}_{\mathrm{stakeholders}} = \int_{\hat{T}\in R} \mathrm{Q@}\hat{T} \times \mathrm{Tolerance}(\hat{T})\, d\hat{T}, \tag{5}$$

where $\mathrm{Q@}\hat{T}$ denotes the quality metric evaluated at time threshold $\hat{T}$, $R = [R_{\min}, R_{\max}]$ is the stakeholder-defined tolerance interval for inference time, and $\mathrm{Tolerance}(\hat{T})$ is a user-defined weighting function that expresses the degree to which inference delays at $\hat{T}$ are acceptable. The integration domain R is determined by the specific requirements of stakeholders, reflecting the time thresholds they consider relevant or acceptable for their application scenarios. For strict time constraints, the tolerance value will be low, reducing the weight of $\mathrm{Q@}\hat{T}$ at those time intervals. On the other hand, for more flexible time requirements, the tolerance value will be higher, increasing the weight of $\mathrm{Q@}\hat{T}$ at those intervals.

Note that Eq. (5) integrates the weighted Q@T across a stakeholder-defined tolerance interval R. It does not represent a normalized average, but rather an aggregate evaluation score that emphasizes performance in regions preferred by the stakeholder. A normalized version can be obtained by dividing by $\int_{\hat{T}\in R} \mathrm{Tolerance}(\hat{T})\, d\hat{T}$, if desired. We intentionally leave the expression unnormalized to maintain flexibility, allowing users to interpret the result either as a total weighted score or to apply normalization as needed for their specific use cases.

This adaptation of Q@T allows us to tailor the evaluation process to the specific time tolerance of the task at hand. In practice, this enables the Q@T evaluation to be much more flexible, reflecting the landscape of stakeholder needs. For example, in autonomous driving, any delay could be catastrophic. Here, Q@T would prioritize faster inference times and impose stricter thresholds. In contrast, in healthcare applications, where accurate diagnosis is critical but minor delays may be acceptable, Q@T would adjust the tolerance to allow for longer inference times while still ensuring high-quality outputs.

By incorporating stakeholder-driven tolerance for inference time into the evaluation process, the Q@T framework can provide a more realistic, adaptable, and comprehensive evaluation metric that reflects the diversity of real-world applications. This adjustment enhances the flexibility of the evaluation, making it more suitable for a variety of use cases where performance criteria differ significantly based on the context and needs of the stakeholders.

Experiments To empirically validate the stakeholder-aware extension of Q@T, we designed two experiments using synthetic tolerance functions. Due to the lack of large-scale data on stakeholder demands, we manually constructed two representative forms of the $\mathrm{Tolerance}(\hat{T})$ function to simulate varying tolerance for inference time. These functions define how much weight each evaluation result $\mathrm{Q@}\hat{T}$ receives at a given inference time threshold $\hat{T}$, thereby reflecting hypothetical stakeholder preferences across different application contexts.

The first form is a linear decay function, controlled by a slope parameter $\alpha \in (0, 1]$:

$$\mathrm{Tolerance}_{\mathrm{l}}(\hat{T}) = 1 - \alpha \cdot \frac{\hat{T} - R_{\min}}{R_{\max} - R_{\min}}. \tag{6}$$

The second form is a nonlinear exponential decay function, controlled by shape parameters $\lambda > 0$ and $p > 1$:

$$\mathrm{Tolerance}_{\mathrm{n}}(\hat{T}) = \exp\left(-\lambda \cdot \left(\frac{\hat{T} - R_{\min}}{R_{\max} - R_{\min}}\right)^{p}\right). \tag{7}$$

In both cases, $R_{\min}$ and $R_{\max}$ define the range of inference time thresholds under consideration. This range represents the domain in which stakeholders are assumed to express meaningful tolerance variations. For instance, in safety-critical scenarios, R may be narrow and focused on low-latency thresholds; in contrast, in offline reasoning tasks, a broader range may apply.

We applied both tolerance functions to two tasks: (1) Emotion recognition using EmotionFlow, evaluated with weighted-F1; and (2)

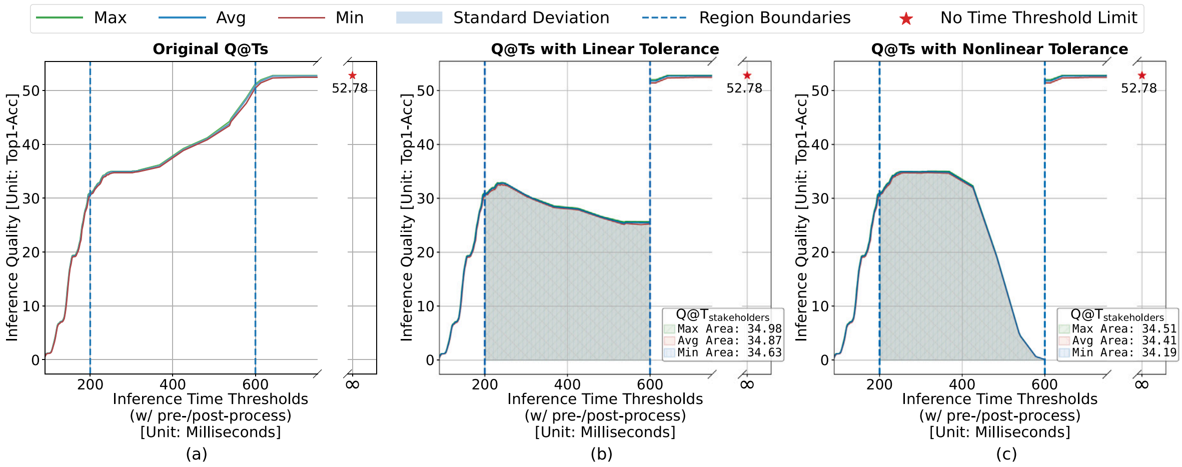


Fig. 7. Stakeholder-aware Q@T evaluation on the Vicuna model (top-1 accuracy). (a) presents the original Q@T curve under varying inference time constraints. (b) depicts the evaluation using a linear tolerance function across the defined interval from 200 ms to 600 ms. (c) applies a nonlinear tolerance function, emphasizing low-latency performance and reducing the impact of slower responses.

Question answering using Vicuna, evaluated with top-1 accuracy. For each case, we compared the original Q@T formulation with the stakeholder-aware variant Q@T$_{\text{stakeholders}}$, using both linear and nonlinear tolerance.

Figs. 6 and 7 demonstrate two key implications of incorporating stakeholder-aware tolerance functions. First, rather than focusing on Q@T under a single time threshold, this framework encourages evaluators to consider a broader range of thresholds, each weighted according to stakeholder preferences. This allows Q@T scores at higher-latency points — often overestimated in unconstrained evaluations — to be reasonably discounted, reflecting more realistic expectations.

Second, the stakeholder-aware score Q@T$_{\text{stakeholders}}$ can be interpreted as a weighted average of Q@T over a tolerance-defined interval R, capturing the system's overall quality across multiple thresholds. This integrates not only peak performance but also performance stability over time, which is crucial for reliable deployment.

Comparing Figs. 6 and 7, we observe that although the nonlinear tolerance function penalizes high-latency Q@T more severely, EmotionFlow exhibits a smoother degradation curve than Vicuna. As a result, its Q@T$_{\text{stakeholders}}$ shows a relative gain under nonlinear weighting. Additionally, Vicuna demonstrates more consistent behavior across the tolerance range, with minimal difference between the maximum area (i.e., best-case Q@T) and the minimum area (i.e., worst-case Q@T) regions—an important indicator of stability in deployment. In contrast, EmotionFlow shows significant performance variance, with a 21.46-point gap in average weighted-F1 between the best and worst Q@T segments.

These results illustrate that the stakeholder-aware Q@T can adapt evaluation outcomes based on context-specific tolerance levels, offering a more flexible and realistic assessment approach. Even with synthetic tolerance profiles, the influence on evaluation is evident, providing preliminary support for the practical applicability of this framework in critical tasks.

5. Conclusion

This paper reanalyzes and extends the Quality@Time-Threshold (Q@T) framework from the perspective of Evaluatology, offering a robust theoretical foundation for evaluating AI inference systems. By applying Evaluatology's five core axioms and its universal evaluation methodology, we redefine the components of the evaluation system, ensuring precise and consistent assessments of AI systems across various tasks and environments. Additionally, we enhance Q@T by incorporating a stakeholder-driven tolerance function, making the framework more adaptable to diverse real-world requirements. This work also emphasizes the importance of tail quality, demonstrating how Q@T captures extreme performance variations overlooked by traditional metrics. Overall, we bridge the gap between theoretical evaluation frameworks and practical AI evaluation, providing a comprehensive, adaptable, and scientifically grounded approach for assessing AI systems in complex applications.

CRediT authorship contribution statement

Zhengxin Yang: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Data curation, Conceptualization..

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E.P. Xing, H. Zhang, J.E. Gonzalez, I. Stoica, Judging LLM-as-a-judge with MT-bench and chatbot arena, 2023, http://dx.doi.org/10.48550/arXiv.2306.05685, arXiv:2306.05685.

[2] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, R. Avila, I. Babuschkin, S. Balaji, V. Balcom, P. Baltescu, H. Bao, M. Bavarian, J. Belgum, I. Bello, J. Berdine, G. Bernadett-Shapiro, C. Berner, L. Bogdonoff, O. Boiko, M. Boyd, A.-L. Brakman, G. Brockman, T. Brooks, M. Brundage, K. Button, T. Cai, R. Campbell, A. Cann, B. Carey, C. Carlson, R. Carmichael, B. Chan, C. Chang, F. Chantzis, D. Chen, S. Chen, R. Chen, J. Chen, M. Chen, B. Chess, C. Cho, C. Chu, H.W. Chung, D. Cummings, J. Currier, Y. Dai, C. Decareaux, T. Degry, N. Deutsch, D. Deville, A. Dhar, D. Dohan, S. Dowling, S. Dunning, A. Ecoffet, A. Eleti, T. Eloundou, D. Farhi, L. Fedus, N. Felix, S.P. Fishman, J. Forte, I. Fulford, L. Gao, E. Georges, C. Gibson, V. Goel, T. Gogineni, G. Goh, R. Gontijo-Lopes, J. Gordon, M. Grafstein, S. Gray, R. Greene, J. Gross, S.S. Gu, Y. Guo, C. Hallacy, J. Han, J. Harris, Y. He, M. Heaton, J. Heidecke, C. Hesse, A. Hickey, W. Hickey, P. Hoeschele, B. Houghton, K. Hsu, S. Hu, X. Hu, J. Huizinga, S. Jain, S. Jain, J. Jang, A. Jiang, R. Jiang, H. Jin, D. Jin, S. Jomoto, B. Jonn, H. Jun, T. Kaftan, Ł. Kaiser, A. Kamali, I. Kanitscheider, N.S. Keskar, T. Khan, L. Kilpatrick, J.W. Kim, C. Kim, Y. Kim, H. Kirchner, J. Kiros, M. Knight, D. Kokotajlo, L.u. Kondraciuk, A. Kondrich, A. Konstantinidis, K. Kosic, G. Krueger, V. Kuo, M. Lampe, I. Lan, T. Lee, J. Leike, J. Leung, D. Levy, C.M. Li, R. Lim, M. Lin, S. Lin, M. Litwin, T. Lopez, R. Lowe, P. Lue, A. Makanju, K. Malfacini, S. Manning, T. Markov, Y. Markovski, B. Martin, K. Mayer, A. Mayne, B. McGrew, S.M. McKinney, C. McLeavey, P. McMillan, J. McNeil, D. Medina, A. Mehta, J. Menick, L. Metz, A. Mishchenko, P. Mishkin, V. Monaco, E. Morikawa, D. Mossing, T. Mu, M. Murati, O. Murk, D. Mély, A. Nair, R. Nakano, R. Nayak, A. Neelakantan, R. Ngo, H. Noh, L. Ouyang, C. O'Keefe, J. Pachocki, A. Paino, J. Palermo, A. Pantuliano, G. Parascandolo, J. Parish, E. Parparita, A. Passos, M. Pavlov, A. Peng, A. Perelman, F.d.B. Peres,

M. Petrov, H.P.d. Pinto, Michael, Pokorny, M. Pokrass, V. Pong, T. Powell, A. Power, B. Power, E. Proehl, R. Puri, A. Radford, J. Rae, A. Ramesh, C. Raymond, F. Real, K. Rimbach, C. Ross, B. Rotsted, H. Roussez, N. Ryder, M. Saltarelli, T. Sanders, S. Santurkar, G. Sastry, H. Schmidt, D. Schnurr, J. Schulman, D. Selsam, K. Sheppard, T. Sherbakov, J. Shieh, S. Shoker, P. Shyam, S. Sidor, E. Sigler, M. Simens, J. Sitkin, K. Slama, I. Sohl, B. Sokolowsky, Y. Song, N. Staudacher, F.P. Such, N. Summers, I. Sutskever, J. Tang, N. Tezak, M. Thompson, P. Tillet, A. Tootoonchian, E. Tseng, P. Tuggle, N. Turley, J. Tworek, J.F.C. Uribe, A. Vallone, A. Vijayvergiya, C. Voss, C. Wainwright, J.J. Wang, A. Wang, B. Wang, J. Ward, J. Wei, C.J. Weinmann, A. Welihinda, P. Welinder, J. Weng, L. Weng, M. Wiethoff, D. Willner, C. Winter, S. Wolrich, H. Wong, L. Workman, S. Wu, J. Wu, M. Wu, K. Xiao, T. Xu, S. Yoo, K. Yu, Q. Yuan, W. Zaremba, R. Zellers, C. Zhang, M. Zhang, S. Zhao, T. Zheng, J. Zhuang, W. Zhuk, B. Zoph, GPT-4 Technical Report, 2023, http://dx.doi.org/10.48550/arXiv.2303.08774, URL http://arxiv.org/abs/2303.08774.

[3] C. Chen, A. Seff, A. Kornhauser, J. Xiao, DeepDriving: Learning affordance for direct perception in autonomous driving, in: 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2722–2730, http://dx.doi.org/10.1109/ICCV.2015.312.

[4] H. Xu, Y. Gao, F. Yu, T. Darrell, End-to-End learning of driving models from large-scale video datasets, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 3530–3538, http://dx.doi.org/10.1109/CVPR.2017.376.

[5] G. Li, Y. Yang, X. Qu, Deep learning approaches on pedestrian detection in hazy weather, IEEE Trans. Ind. Electron. 67 (10) (2020) 8889–8899, http://dx.doi.org/10.1109/TIE.2019.2945295.

[6] S.M. McKinney, M. Sieniek, V. Godbole, J. Godwin, N. Antropova, H. Ashrafian, T. Back, M. Chesus, G.S. Corrado, A. Darzi, M. Etemadi, F. Garcia-Vicente, F.J. Gilbert, M. Halling-Brown, D. Hassabis, S. Jansen, A. Karthikesalingam, C.J. Kelly, D. King, J.R. Ledsam, D. Melnick, H. Mostofi, L. Peng, J.J. Reicher, B. Romera-Paredes, R. Sidebottom, M. Suleyman, D. Tse, K.C. Young, J. De Fauw, S. Shetty, Addendum: International evaluation of an AI system for breast cancer screening, Nature 586 (7829) (2020) http://dx.doi.org/10.1038/s41586-020-2679-9, E19–E19. URL https://www.nature.com/articles/s41586-020-2679-9.

[7] P. Rajpurkar, J. Irvin, R.L. Ball, K. Zhu, B. Yang, H. Mehta, T. Duan, D. Ding, A. Bagul, C.P. Langlotz, B.N. Patel, K.W. Yeom, K. Shpanskaya, F.G. Blankenberg, J. Seekins, T.J. Amrhein, D.A. Mong, S.S. Halabi, E.J. Zucker, A.Y. Ng, M.P. Lungren, Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists, PLOS Med. 15 (11) (2018) e1002686, http://dx.doi.org/10.1371/journal.pmed.1002686, URL https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1002686.

[8] M.F. Ijaz, G. Alfian, M. Syafrudin, J. Rhee, Hybrid prediction model for Type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest, Appl. Sci. 8 (8) (2018) 1325, http://dx.doi.org/10.3390/app8081325, URL https://www.mdpi.com/2076-3417/8/8/1325.

[9] Y. Zhang, A. Carballo, H. Yang, K. Takeda, Autonomous driving in adverse weather conditions: A survey, 2021, CoRR, arXiv:2112.08936.

[10] D. Wang, W. Fu, Q. Song, J. Zhou, Potential risk assessment for safe driving of autonomous vehicles under occluded vision, Sci. Rep. 12 (2022) http://dx.doi.org/10.1038/s41598-022-08810-z, URL https://api.semanticscholar.org/CorpusID:247628492.

[11] Z. Yang, W. Gao, C. Luo, L. Wang, F. Tang, X. Wen, J. Zhan, Quality at the tail of machine learning inference, 2024, arXiv:2212.13925.

[12] W. Gao, L. Wang, M. Chen, J. Xiong, C. Luo, W. Zhang, Y. Huang, W. Li, G. Kang, C. Zheng, B. Xie, S. Dai, Q. He, H. Ye, Y. Bao, J. Zhan, High fusion computers: The IoTs, edges, data centers, and humans-in-the-loop as a computer, BenchCouncil Trans. Benchmarks, Stand. Eval. 2 (3) (2022) 100075, http://dx.doi.org/10.1016/j.tbench.2022.100075, URL https://www.sciencedirect.com/science/article/pii/S277248592200062X.

[13] F.-L. Fan, J. Xiong, M. Li, G. Wang, On interpretability of artificial neural networks: A survey, IEEE Trans. Radiat. Plasma Med. Sci. 5 (6) (2021) 741–760, http://dx.doi.org/10.1109/TRPMS.2021.3066428.

[14] Y. Zhang, P. Tiňo, A. Leonardis, K. Tang, A survey on neural network interpretability, IEEE Trans. Emerg. Top. Comput. Intell. 5 (5) (2021) 726–742, http://dx.doi.org/10.1109/TETCI.2021.3100641.

[15] V.J. Reddi, C. Cheng, D. Kanter, P. Mattson, G. Schmuelling, C.-J. Wu, B. Anderson, M. Breughe, M. Charlebois, W. Chou, R. Chukka, C. Coleman, S. Davis, P. Deng, G. Diamos, J. Duke, D. Fick, J.S. Gardner, I. Hubara, S. Idgunji, T.B. Jablin, J. Jiao, T.S. John, P. Kanwar, D. Lee, J. Liao, A. Lokhmotov, F. Massa, P. Meng, P. Micikevicius, C. Osborne, G. Pekhimenko, A.T.R. Rajan, D. Sequeira, A. Sirasao, F. Sun, H. Tang, M. Thomson, F. Wei, E. Wu, L. Xu, K. Yamada, B. Yu, G. Yuan, A. Zhong, P. Zhang, Y. Zhou, MLPerf inference benchmark, in: 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), 2020, pp. 446–459, http://dx.doi.org/10.1109/ISCA45697.2020.00045.

[16] W. Gao, F. Tang, L. Wang, J. Zhan, C. Lan, C. Luo, Y. Huang, C. Zheng, J. Dai, Z. Cao, D. Zheng, H. Tang, K. Zhan, B. Wang, D. Kong, T. Wu, M. Yu, C. Tan, H. Li, X. Tian, Y. Li, J. Shao, Z. Wang, X. Wang, H. Ye, AIBench: An Industry Standard Internet Service AI Benchmark Suite, 2019, http://dx.doi.org/10.48550/arXiv.1908.08998, arXiv:1908.08998.

[17] W. Gao, C. Luo, L. Wang, X. Xiong, J. Chen, T. Hao, Z. Jiang, F. Fan, M. Du, Y. Huang, F. Zhang, X. Wen, C. Zheng, X. He, J. Dai, H. Ye, Z. Cao, Z. Jia, K. Zhan, H. Tang, D. Zheng, B. Xie, W. Li, X. Wang, J. Zhan, AIBench: Towards scalable and comprehensive datacenter AI benchmarking, in: C. Zheng, J. Zhan (Eds.), Benchmarking, Measuring, and Optimizing, in: Lecture Notes in Computer Science, Springer International Publishing, Cham, 2019, pp. 3–9, http://dx.doi.org/10.1007/978-3-030-32813-9_1.

[18] C. Coleman, D. Narayanan, D. Kang, T. Zhao, J. Zhang, L. Nardi, P. Bailis, K. Olukotun, C. Ré, M. Zaharia, DAWNBench: An end-to-end deep learning benchmark and competition, in: Workshop on ML Systems At Advances in Neural Information Processing Systems, 2017.

[19] J. Zhan, L. Wang, W. Gao, H. Li, C. Wang, Y. Huang, Y. Li, Z. Yang, G. Kang, C. Luo, H. Ye, S. Dai, Z. Zhang, Evaluatology: The science and engineering of evaluation, BenchCouncil Trans. Benchmarks, Stand. Eval. 4 (1) (2024) 100162, http://dx.doi.org/10.1016/j.tbench.2024.100162.

[20] J. Zhan, Five axioms of things, BenchCouncil Trans. Benchmarks, Stand. Eval. 4 (3) (2024) 100184, http://dx.doi.org/10.1016/j.tbench.2024.100184.

[21] J. Zhan, A short summary of evaluatology: The science and engineering of evaluation, BenchCouncil Trans. Benchmarks, Stand. Eval. 4 (2) (2024) 100175, http://dx.doi.org/10.1016/j.tbench.2024.100175.

[22] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, LightGCN: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 639–648, http://dx.doi.org/10.1145/3397271.3401063.

[23] D. Ayata, Y. Yaslan, M.E. Kamasak, Emotion recognition from multimodal physiological signals for emotion aware healthcare systems, J. Med. Biological Eng. 40 (2) (2020-04-01) 149–157, http://dx.doi.org/10.1007/s40846-019-00505-7.

[24] S. Zepf, J. Hernandez, A. Schmitt, W. Minker, R.W. Picard, Driver emotion recognition for intelligent vehicles: A survey, ACM Comput. Surv. 53 (3) (2020-07-04) 64:1–64:30, http://dx.doi.org/10.1145/3388790, URL https://dl.acm.org/doi/10.1145/3388790.

[25] M. Dhuheir, A. Albaseer, E. Baccour, A. Erbad, M. Abdallah, M. Hamdi, Emotion recognition for healthcare surveillance systems using neural networks: A survey, in: 2021 International Wireless Communications and Mobile Computing (IWCMC), 2021-06, pp. 681–687, http://dx.doi.org/10.1109/IWCMC51323.2021.9498861.

[26] T. Turay, T. Vladimirova, Toward performing image classification and object detection with convolutional neural networks in autonomous driving systems: A survey, IEEE Access 10 (2022) 14076–14119, http://dx.doi.org/10.1109/ACCESS.2022.3147495.

[27] E. Yurtsever, J. Lambert, A. Carballo, K. Takeda, A survey of autonomous driving: Common practices and emerging technologies, IEEE Access 8 (2020) 58443–58469, http://dx.doi.org/10.1109/ACCESS.2020.2983149.

[28] Q. Guo, S. Chen, X. Xie, L. Ma, Q. Hu, H. Liu, Y. Liu, J. Zhao, X. Li, An empirical study towards characterizing deep learning development and deployment across different frameworks and platforms, in: 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), 2019, pp. 810–822, http://dx.doi.org/10.1109/ASE.2019.00080.

[29] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, URL https://www.tensorflow.org/ Software available from tensorflow.org.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, 2019, http://dx.doi.org/10.48550/arXiv.1912.01703, arXiv:1912.01703.

[31] Y. Wu, L. Liu, C. Pu, W. Cao, S. Sahin, W. Wei, Q. Zhang, A comparative measurement study of deep learning as a service framework, IEEE Trans. Serv. Comput. 15 (1) (2022-01) 551–566, http://dx.doi.org/10.1109/TSC.2019.2928551.

[32] J. Zhan, A BenchCouncil view on benchmarking emerging and future computing, BenchCouncil Trans. Benchmarks, Stand. Eval. 2 (2) (2022) 100064, http://dx.doi.org/10.1016/j.tbench.2022.100064, URL https://www.sciencedirect.com/science/article/pii/S2772485922000515.

[33] X. Song, L. Zang, R. Zhang, S. Hu, L. Huang, Emotionflow: Capture the dialogue level emotion transitions, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 8542–8546, http://dx.doi.org/10.1109/ICASSP43922.2022.9746464.