

From Efficiency to Sufficiency: The Cost-Aware Measurement Design Framework for Automated Essay Scoring

Yi Gui

Measurement Incorporated

ygui@measinc.com

Abstract

Compute cost in automated essay scoring (AES) is usually treated as a technical efficiency issue rather than a measurement design concern. Yet deployment decisions involve more than latency, model size, or inference expense: in high-stakes settings total burden includes validation, monitoring, adjudication, compliance, and governance, while in instructional settings pricing, licensing, and integration may dominate. This paper reframes AES as a cost-constrained measurement system and introduces the Cost-Aware Measurement Design (CAMD) Framework, a four-layer structure—metering, allocation, unit economics, and decision—that translates raw computational signals into decision-relevant cost constructs. It defines cost per defensible score (CPDS) and argues that AES systems should be compared by quality-cost sufficiency: minimizing cost among designs that satisfy measurement requirements. Because scoring volume conditions all four layers, CPDS is a function of scale rather than a single transferable number.

1 Introduction

Automated essay scoring (AES) research has long emphasized efficiency: reduced human scoring burden, faster turnaround, greater scalability. Yet the field lacks a stable language for representing compute cost beyond proxies such as speed, model size, or reduced labor, so efficiency claims are difficult to compare, transfer, or interpret within measurement-oriented decision making (Dikli, 2006; Bennett, 2004; Rotou et al., 2020).

This limitation is increasingly consequential. In high-stakes programs the scoring engine is one component within a broader system of psychometric validation, security, and compliance, where cloud inference may be a minor share of total cost relative to adjudication, validity studies, auditability, and governance. In instructional and formative contexts, cost structures resemble SaaS

models, where licensing, API pricing, and platform integration dominate (Educational Testing Service, 2021; Fang et al., 2025; Bai & Hui, 2024). Cost behaves as a system-level construct rather than a model-level metric.

The problem is not that AES ignores cost, but that it lacks a way of representing cost that is portable, comparable, and decision-relevant. In cloud operations and machine learning systems, cost is routinely metered, allocated, and normalized into units that support decisions. AES studies still rely on isolated indicators that do not distinguish direct from shared burden, total from unit cost, or technically cheap from operationally sufficient (FinOps Foundation, n.d.-a, n.d.-c; OpenCost Specification, n.d.; Li et al., 2023). Cost-Aware Measurement Design (CAMD) supplies the missing framework: it represents cost through four cumulative layers, defines cost per defensible score (CPDS) as its central unit, and compares scoring designs under quality-cost sufficiency rather than by raw performance.

2 The AES Gap: Efficiency Without Decision-Relevant Cost Representation

Existing AES literature has not failed to recognize efficiency, but has failed to represent it in a portable, comparable, decision-relevant form. Early reviews frame automated scoring as a response to the labor, time, and financial burden of hand scoring; operational guidance emphasizes monitoring, validation, and quality control; measurement theory treats cost, feasibility, and utility as legitimate dimensions of assessment design; educational cost analysis supplies ingredients-based costing; and efficient NLP and Green AI have normalized reporting computational burden rather than leaving it invisible. Missing is their integration into a portable compute-cost framework for modern AES (Dikli, 2006; Rotou et al., 2020; van der Vleuten, 1996; Levin et al., 2018; Institute of Education Science

es, 2020; Schwartz et al., 2020; Tay et al., 2022). Table 1 summarizes these strands and the gap each leaves unresolved.

The dominant reporting vocabulary centers on score agreement, latency, model size, and sometimes reduced human effort. Those metrics are useful, but they do not constitute a cost representation language: they rarely specify the boundary of cost, how shared or idle burden is treated, or which unit of output should anchor comparison (Dikli, 2006; Rupp et al., 2019). Table 2 contrasts common AES efficiency reporting with the elements of a structured cost representation.

Current reporting therefore lacks three properties required for cost reasoning: an explicit cost boundary, a principled allocation of shared and indirect burden, and a decision-relevant unit of analysis. The limitation of prior approaches is not simply omission but representational mismatch: existing metrics describe computational properties, whereas AES requires cost constructs defined at the level of defensible scoring outcomes.

3 Motivation and Contributions

Although conceptual in form, the framework is motivated by recurring patterns in AES research: substantial reductions in representation dimensionality often yield minimal degradation in scoring performance; variability from stochastic training factors such as random seeds can exceed the gains from architectural modification; and design choices that appear efficient at the model level can raise downstream operational burden through instability and review requirements.

These patterns point to a structural limitation. In AES, cost cannot be treated as an unconstrained optimization objective: scoring systems must first satisfy minimum requirements for validity, reliability, and fairness before cost becomes meaningful. The relevant problem is not how to minimize computational expense in isolation, but how to reason about cost within a constrained measurement space.

This paper makes three contributions. First, it reframes AES as a cost-constrained measurement system rather than treating cost as an external engineering consideration. Second, it introduces CAMD, a four-layer framework—metering, allocation, unit economics, and decision—that transforms raw computational signals into decision-

relevant cost constructs. Third, it defines CPDS and formalizes quality-cost sufficiency as the basis for comparing AES designs. The contribution lies not in inventing cost-accounting machinery but in translating mature tools from cloud cost management, carbon accounting, and machine learning systems research into a measurement-centered framework.

4 Defining the Cost Boundary

An initial framework should not model every possible burden at once; doing so would blur the analytic target. This paper therefore scopes direct operational cost: the resources required to run scoring under deployment or near-deployment conditions, including inference compute, attributable storage, orchestration, monitoring, and reserved capacity reasonably assigned to the scoring workload (Levin et al., 2018; Institute of Education Sciences, 2020; OpenCost Specification, n.d.). At minimum the boundary should include model serving, prompt and context burden, batchability, cacheability, storage and audit retention, human review volume attributable to machine uncertainty, and validation overhead, because the model call is often only one ingredient, and not always the most expensive (Educational Testing Service, 2021; Rotou et al., 2020).

The boundary is intentionally narrower than the universe of all AI-related costs. Training cost may matter when models are retrained or substantially fine-tuned; carbon and energy reporting matter for sustainability and some institutional decisions; embodied infrastructure matters for full life-cycle accounting. These are best treated as secondary extensions in a first conceptual statement; their role here is analogical, showing that mature fields already reason about boundaries, indirect burden, amortization, and shared infrastructure (Schwartz et al., 2020; Patterson et al., 2021; Lannelongue et al., 2021; CarbonTracker, n.d.; Google Cloud, n.d.).

Constraining the boundary improves both tractability and portability: researchers can begin with a limited ingredients-style model of operational burden without full billing exports or machine-level telemetry, and institutions differing in pricing and scale can still share a schema for which components are counted and how they combine (FOCUS Specification, n.d.; Institute of Education Sciences, 2020).

5 External Literatures and Emerging AES Evidence

CAMD draws on several external literatures, but their role is deliberately bounded; Table 3 translates them into AES design language. FinOps contributes unit economics and the distinction between resource-side and workload-side metrics. FOCUS contributes the idea that portability can exist at the level of schema and accounting semantics even when prices differ. OpenCost contributes an explicit logic for shared, idle, and overhead burden. Green AI and carbon accounting contribute boundary discipline and the norm that computational burden be reportable. Machine learning systems research contributes the insight that cost belongs in the design objective itself (FinOps Foundation, n.d.-a, n.d.-b, n.d.-c; FOCUS Specification, n.d.; OpenCost Specification, n.d.; Schwartz et al., 2020; Li et al., 2023).

Emerging AES-specific evidence, though thinner, points the same way. Direct cost analyses in writing screening suggest automation becomes economically stronger as assessment volume and frequency increase. Work on incremental learning and compact multi-task architectures suggests retraining cadence, storage, and maintenance debt are themselves cost drivers. Comparisons of traditional and LLM-based scoring suggest higher inference cost does not automatically buy better scoring performance, even when it buys richer feedback. AES requires a system-level cost language, not merely a faster model (Matta et al., 2022; Bai & Hui, 2024; Fang et al., 2025).

6 The CAMD Framework

CAMD has four cumulative layers: metering, allocation, unit economics, and decision. Cost must first be observed, then assigned, then normalized, and only then used to support design decisions. The framework is measurement-informed because its decision layer is anchored in validity, reliability, fairness, feasibility, and utility. Scoring volume, denoted N , runs through all four layers rather than forming one of its own: it is metered as an observable, governs how fixed and variable components are allocated, conditions the unit in which cost is expressed, and shifts the thresholds against which sufficiency is judged.

6.1 Metering

The metering layer records the observables that make cost analysis possible. The first is the scoring volume N , together with the deployment regime: workstation CPU, a single accelerator, a managed cluster, or a third-party API. Neither is incidental metadata. Every resource-side quantity is interpretable only relative to them, because the same algorithm yields very different totals at $N = 100$ and at $N = 1,000,000$, and because the regime itself changes discontinuously as N grows. A minimal AES set would include GPU- or CPU-hours, memory use, storage footprint, throughput, latency, queue time, request volume, and configuration details such as model size, prompt length, or compression setting. Developers need not meter everything; they should identify a stable core of observables reusable across studies (MLCommons, n.d.; Lannelongue et al., 2021).

6.2 Allocation

Metering alone does not produce interpretable comparison. AES workloads may run in shared environments, reserve capacity beyond their realized use, or depend on orchestration and monitoring that do not appear in a narrow inference window. The allocation layer specifies which components are direct, shared, idle, or reserved, and how each is attributed. It must also separate cost that is fixed with respect to N from cost that varies with it: environment preparation, validation studies, and reserved capacity are incurred once and spread across the essays scored, whereas per-inference compute, expected human review, and per-essay storage grow with N . Reporting the two separately is what makes a cost claim portable: a reader under similar hardware and workflow recovers their own per-essay figure by dividing the fixed component by their N and adding the variable one. A blended figure cannot be transferred, because it silently encodes the volume at which it was produced (OpenCost Specification, n.d.; Li et al., 2023).

6.3 Unit Economics and Cost per Defensible Score

Allocated components must then be normalized into units that matter for operational and assessment decisions: cost per essay, per 1,000 essays, per batch, per scoring day, or per configuration

that reaches an agreed level of measurement quality. Raw technical signals communicate poorly with program administrators, procurement stakeholders, and assessment designers; unit economics turns engineering burden into operationally interpretable burden (FinOps Foundation, n.d.-c).

Every such unit is conditional on N . Because fixed cost is spread over the scored volume while monitoring, adjudication, and governance need not grow proportionally with it, cost per essay is a function of N rather than a constant, commonly falling as fixed cost is absorbed and not necessarily monotonic thereafter. CPDS should therefore be reported as $\text{CPDS}(N)$, with the volume and regime at which it was obtained. A single value is evidence about one point on that curve; it does not license comparison with a value obtained at a different scale.

This has a practical consequence. Where feasible, cost should be reported at more than one volume, for example the full corpus and a subsample an order of magnitude smaller. Within a single regime, two volumes let a reader recover the implied fixed and variable components, since the difference in total cost between the runs is attributable to the variable component alone, and check that recovery against the authors' decomposition. The check assumes cost is approximately linear in N across the interval spanned, which fails wherever the regime changes; disagreement is therefore informative. Formally, CPDS is the expected total cost of a scoring run divided by the expected number of scores it delivers:

$$\text{CPDS}(d, N) = \frac{E[C_{\text{total}}(d, N)]}{E[S(d, N)]}$$

The numerator is the expected total cost of scoring N responses under configuration d ; the denominator, $S(d, N)$, is the expected number of scores that configuration delivers for use. S equals N when every response completes the workflow, and is smaller when responses are routed out as unscorable, left unresolved in adjudication, or withdrawn on appeal: their cost is incurred without yielding a usable score, and the ratio makes that visible. Throughout, d is restricted to configurations satisfying the sufficiency thresholds of the decision layer.

A defensible score is shorthand for a score produced under a configuration shown, for a de-

clared interpretation and use, to satisfy predeclared thresholds on validity-relevant evidence, reliability, fairness, and operational adequacy. Validity is a property of the interpretation and use of scores rather than of scores or instruments themselves, so a score cannot be valid in isolation (Messick, 1989; Kane, 2013; AERA, APA, & NCME, 2014). CPDS prices the conditions under which a score can be defended for a stated use; it does not treat defensibility as a property the score carries on its own.

CPDS challenges the assumption that cost per inference is the relevant economic unit. Operationally, a model output is not yet a defensible score: it may still trigger adjudication, monitoring, audit logging, error correction, appeal handling, or subgroup review (Rotou et al., 2020; Educational Testing Service, 2021; Matta et al., 2022). Lower raw inference cost therefore need not imply lower effective cost. Table 4 compares two configurations that differ in both inference cost and downstream burden, at the same volume and under a common regime, without which their CPDS values would not be comparable. Configuration A has lower per-inference cost but less stable scores and a higher human review rate; Configuration B has higher inference cost, more consistent outputs, and less downstream intervention.

Despite its lower per-inference cost, Configuration A yields the higher cost per defensible score once downstream review is counted, whereas Configuration B achieves the lower CPDS by reducing intervention. Design choices redistribute cost across components: cost is a property not of the model alone but of the full scoring process required to produce a defensible outcome. CPDS shifts the unit of cost from computation to defensibility—the question is not what it costs to generate a score, but what it costs to produce one that can be defended.

6.4 Decision

The final layer is where compute cost becomes explicitly psychometric. AES systems are not chosen because one model has a slightly better score correlation or lower latency, but because they are adequate for an intended use under realistic constraints. Cost comparison should therefore run under sufficiency thresholds rather than unrestricted performance ranking (van der Vleu-

ten, 1996; Schuwirth & van der Vleuten, 2011). Let

$$D^*(N) = \{d \in D: V(d) \geq v^*, R(d) \geq r^*, F(d) \geq f^*, O(d, N) \geq o^*\},$$

and then choose

$$d^*(N) = \arg \min_{d \in D^*(N)} \text{CPDS}(d, N).$$

Here V , R , and F represent validity-relevant, reliability-related, and fairness-related evidence; O represents operational sufficiency, including timeliness and feasibility; and $\text{CPDS}(d, N)$ is defined only once those thresholds and the operating scale are declared. Sufficiency is itself scale-dependent, and the operational threshold in particular is a function of N : a design routing every essay to a human reader may satisfy it at $N = 50$ and fail outright at $N = 500,000$, while a configuration provisioned for continuous high-volume scoring is wasteful at small N without being more defensible. The framework does not identify a best AES system; it identifies the least costly sufficient design at a declared operating scale.

The formulation is compatible with generalizability-theory-based design optimization, where reliability targets act as constraints while alternative configurations—number of models, ensemble size, rescoring passes, human-review triggers—imply different cost profiles (Parkes, 2000; Jiang et al., 2021). The parallel runs deeper than vocabulary. A generalizability study estimates variance components relative to a declared universe of admissible observations, and a decision study projects reliability at whatever number of raters or tasks a program can afford: the coefficient from one design does not transfer, but the components generating it do (Brennan, 2001). Cost behaves the same way. Components recovered at two or more volumes support projection under comparable hardware and workflow, whereas CPDS at a single N does not.

7 Measurement Principles Underlying CAMD

CAMD embeds cost in the logic of assessment design across five measurement dimensions.

1. **Validity.** Cost does not override intended score interpretation and use. A cheaper system is not preferable if the evidence base for intended inferences becomes weaker.

2. **Reliability.** A lower-cost design is attractive only if it maintains acceptable consistency, agreement, or generalizability for the scoring purpose.
3. **Fairness.** Cost-saving choices should be evaluated for their impact on subgroup performance, bias, and differential degradation across populations.
4. **Feasibility.** Time-to-score, staffing realities, monitoring burden, and infrastructure constraints are part of the operational adequacy of a design.
5. **Utility.** The desirable design is rarely the best on one dimension; it is the one that best supports intended use under realistic trade-offs.

This psychometric anchoring changes the interpretation of cost. Cost is not a separate optimization problem sitting beside measurement quality; it is one of the design dimensions through which assessment systems become more or less defensible. Cost-aware AES is thus a natural extension of long-standing assessment utility arguments, not a foreign import from cloud engineering (van der Vleuten, 1996; Parkes, 2000).

8 A Research Agenda for Cost-Aware Measurement Design

The framework also opens a broader agenda for cost-aware measurement design.

Empirical CPDS estimation across AES architectures. Traditional AES, transformer-based AES, and LLM-based scoring can be compared not only by agreement or validity evidence, but by the expected cost of producing a defensible score.

Cost-aware G-studies and D-studies. Reliability or dependability targets can be treated as constraints while alternative scoring designs imply different cost profiles.

Cost $\times$ fairness $\times$ decision-risk analysis. Future work can examine whether apparently cheaper systems shift burden onto subgroup disparities, appeals, or human override processes.

Cost-aware ensemble and hybrid-design selection. The question is not whether one scorer is globally best, but whether a hybrid, ensemble, or fallback design reduces CPDS in a given deployment context.

Deployment-specific cost modeling. High-stakes assessment, K-12 screening, classroom

practice, formative revision, and tutoring should not be assumed to share a cost structure or the same defensibility thresholds.

Cost reporting protocols and their validation. A reporting protocol specifies the fields a study must disclose rather than the corpus it must run: scoring volume, deployment regime, batch configuration, the boundary of what is counted, fixed and variable components reported separately, human review rate, and the thresholds under which CPDS was computed. Work is needed both to fix that field set and to test it, by having independent programs describe the same system under the protocol and examining whether their cost claims become mutually recomputable.

Public cost benchmarks, and what must precede them. A shared corpus and reference deployment configuration would allow the third-party verification that self-report cannot provide. Existing corpora supply standardized essay inputs but not scoring volume, deployment regime, or the adjudication, monitoring, and governance processes that dominate operational cost, so the dominant components would have to be simulated rather than measured. Because a corpus can be subsampled but not extended, the largest observable volume is bounded by corpus size, and public AES corpora are small relative to operational volumes. Protocol development must therefore come first.

In the near term, the framework supports immediate reporting reform. AES studies can move beyond isolated efficiency statements by reporting a minimum set of direct operational ingredients, an explicit cost boundary, and at least one interpretable unit-economic metric, each accompanied by the volume and regime at which it was obtained. Standardizing components and formulas rather than prices supports cumulative comparison across institutions. Once that layer is established, the framework can be extended to broader human labor, retraining schedules, carbon reporting, or amortized infrastructure cost.

9 Limitations

This framework is subject to several limitations. First, it prioritizes direct operational cost and does not fully incorporate training, carbon, or long-term infrastructure considerations. Second, CPDS depends on validity, reliability, and fairness thresholds that vary across programs and

require further standardization. Third, it is most directly applicable to medium- to large-scale AES systems; small-scale or low-resource settings may require adaptation. Fourth, the reporting requirements are not themselves costless. Disclosing volume, regime, and a fixed-variable decomposition imposes a low incremental burden where studies already collect run-level telemetry, but mandating a common benchmark imposes a positive and recurring one, since the benchmark run is additional to the work a program needs and must be repeated as hardware and pricing change. That burden falls unevenly: a fixed compliance cost is heaviest for those spreading it over the smallest volume, the same arithmetic the framework applies to scoring cost.

10 Conclusion

The central claim of this paper is not that AES has overlooked efficiency, but that it still lacks a structured and psychometrically anchored language for discussing cost. That is why the field continues to oscillate between broad efficiency narratives and narrow engineering proxies. CAMD is offered as a needed conceptual advance before another round of empirical comparison can fully pay off.

Reframing AES as a cost-constrained measurement system places cost inside the logic of assessment design, models compute burden as a multi-layer construct, and insists that comparisons be made under quality-cost sufficiency rather than raw performance rankings. CAMD gives testing programs and vendors a common vocabulary for metering and reporting the cost of scoring; it gives researchers a reporting standard that makes cost claims comparable across studies and hardware environments; and it gives score users a basis for asking whether cost-saving choices have compromised validity, reliability, or fairness.

References

- American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). *Standards for educational and psychological testing*. American Educational Research Association.
- Bai, H., & Hui, S. C. (2024). A crowdsourcing-based incremental learning framework for automated essays scoring. *Expert Systems with Applications*, 238(Part B), 121755.
- Bennett, R. E. (2004). Moving the field forward: Some thoughts on validity and automated scoring. *Journal of Technology, Learning, and Assessment*, 3(4), 1-15.
- Brennan, R. L. (2001). *Generalizability theory*. Springer-Verlag.
- CarbonTracker. (n.d.). <https://carbontracker.info/>
- Dikli, S. (2006). An overview of automated scoring of essays. *Journal of Technology, Learning, and Assessment*, 5(1), 1-35.
- Educational Testing Service. (2021). Best practices for constructed-response scoring. https://www.ets.org/pdfs/about/cr_best_practices.pdf
- Fang, L., Wang, T., Ma, P., & Zhai, X. (2025). Generalizable and efficient automated scoring with a knowledge-distilled multi-task mixture-of-experts. arXiv:2511.17601. <https://arxiv.org/abs/2511.17601>
- FinOps Foundation. (n.d.-a). FinOps for AI. <https://www.finops.org/framework/technology-categories/ai/>
- FinOps Foundation. (n.d.-b). GenAI FinOps: How Token Pricing Really Works. <https://www.finops.org/wg/genai-finops-how-token-pricing-really-works/>
- FinOps Foundation. (n.d.-c). Unit Economics. <https://www.finops.org/framework/capabilities/unit-economics/>
- FOCUS Specification. (n.d.). <https://focus.finops.org/focus-specification/>
- Google Cloud. (n.d.). Carbon Footprint Reporting Methodology. <https://docs.cloud.google.com/carbon-footprint/docs/methodology>
- Institute of Education Sciences. (2020). Cost analysis: A starter kit (IES 2020-001). U.S. Department of Education. <https://ies.ed.gov/sites/default/files/rel-cen-tral/document/2024/10/Cost%20Analysis%20A%20Starter%20Kit.pdf>
- Jiang, S., Shi, D., & DiStefano, C. (2021). Using machine learning to optimize cost and generalizability in assessment design. *Educational and Psychological Measurement*, 81(4), 657-682.
- Kane, M. T. (2013). Validating the interpretations and uses of test scores. *Journal of Educational Measurement*, 50(1), 1-73.
- Lannelongue, L., Grealey, J., & Inouye, M. (2021). Green algorithms: Quantifying the carbon footprint of computation. *Advanced Science*, 8(12), 2100707. <https://doi.org/10.1002/advs.202100707>
- Levin, H. M., McEwan, P. J., Belfield, C., Bowden, A., & Shand, R. (2018). *Economic evaluation in education: Cost-effectiveness and benefit-cost analysis* (2nd ed.). SAGE.
- Li, Z., Zheng, L., Zhong, Y., Liu, V., Sheng, Y., Jin, X., Huang, Y., Chen, Z., Zhang, H., Gonzalez, J. E., & Stoica, I. (2023). AlpaServe: Statistical multiplexing with model parallelism for deep learning serving. <https://www.usenix.org/conference/osdi23/presentation/li-zhouhan>
- Matta, T., Keller-Margulis, M., & Mercer, S. H. (2022). Cost analysis and cost-effectiveness of hand-scored and automated writing screening approaches. *Journal of School Psychology*, 91, 1-17.
- Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Educational measurement* (3rd ed., pp. 13-103). American Council on Education/Macmillan.
- MLCommons. (n.d.). MLPerf Inference Power Measurement. <https://docs.mlcommons.org/inference/power/>
- OpenCost Specification. (n.d.). <https://opencost.io/docs/specification/>
- Parkes, J. (2000). The relationship between reliability and cost of performance assessments. *Educational Measurement: Issues and Practice*, 19(3), 5-13.
- Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., & Dean, J. (2021). Carbon emissions and large neural network training. <https://arxiv.org/abs/2104.10350>
- Rotou, O., Foltz, P. W., & Pearson, P. D. (2020). Automated scoring: The path to operational deployment. *Educational Measurement: Issues and Practice*, 39(3), 30-38.

- Rupp, A. A., Ferne, T., & Choi, H. (2019). How automated scoring systems can be used to evaluate constructed responses. *Educational Measurement: Issues and Practice*, 38(3), 45-55.
- Schuwirth, L. W. T., & van der Vleuten, C. P. M. (2011). Programmatic assessment: From assessment of learning to assessment for learning. *Medical Teacher*, 33(6), 478-485.
- Schwartz, R., Dodge, J., Smith, N. A., & Etzioni, O. (2020). Green AI. *Communications of the ACM*, 63(12), 54-63.
- Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. *ACM Computing Surveys*, 55(6), 1-28.
- van der Vleuten, C. P. M. (1996). The assessment of professional competence: Developments, research and practical implications. *Advances in Health Sciences Education*, 1(1), 41-67.

Table 1: Representative strands relevant to AES cost analysis.

Literature strand	Main contribution	What remains missing for AES
Early and general AES literature	Recognizes labor reduction, faster scoring, and operational scale as core motivations for automation.	Does not formalize compute cost as a structured construct or provide portable reporting rules for contemporary model deployments.
AES deployment guidance	Emphasizes validation, monitoring, fairness, and quality control in operational use.	Usually treats cost as a background consideration rather than a design variable with explicit boundaries and allocation rules.
Psychometric and assessment-utility theory	Shows that assessment quality must be considered alongside feasibility, cost, and use.	Does not yet specify how compute-related burden should be represented in transformer-based scoring systems.
Educational cost analysis	Provides ingredients-based costing and cost-effectiveness logic.	Has seen limited extension to modern AES deployments where cost depends on model serving, storage, orchestration, and reserved capacity.
Efficient NLP and Green AI	Normalizes reporting of latency, energy, memory, and efficiency trade-offs.	Usually evaluates burden at the model or system level without grounding comparisons in measurement sufficiency for assessment use.

Table 2: Stylized gap demonstration: common AES efficiency reporting versus structured cost representation.

Often reported or discussed	Rarely made explicit	Why the omission matters
Model size, runtime, latency, throughput, or reduced human scoring time	Formal cost boundary	Without a boundary, it is unclear whether the claim concerns direct inference only or a broader operational workflow.
Aggregate performance and efficiency claims	Allocation rules for shared, idle, or overhead cost	Operationally meaningful comparison requires specifying what portion of infrastructure burden belongs to the scoring workload.
Technical metrics for a specific experiment	Portable unit economics such as cost per essay, cost per defensible score, or cost per acceptable configuration	Stakeholders need interpretable workload units rather than raw engineering signals alone.
Accuracy-oriented ranking of models	Decision rules linking cost to validity, reliability, fairness, and feasibility thresholds	Measurement design rarely chooses systems by score quality alone; sufficiency under intended use is the actual decision problem.

Table 3: Translating mature external methods into AES design language.

External literature	Transferable concept	AES translation
FinOps and FinOps for AI	Unit economics; resource-side versus service-side metrics	Distinguish cost per GPU-hour from cost per essay or cost per acceptable-scoring configuration.
FOCUS	Portable schema and common cost semantics	Standardize cost components and formulas across AES studies even when dollar values differ by institution.
OpenCost	Allocation of direct, shared, idle, and overhead burden	Represent not only inference cost but also reserved capacity, monitoring, storage, and shared operational support.
Carbon-accounting frameworks	Boundary discipline; treatment of indirect and amortized burden	Clarify what is included in operational AES cost and what is treated as a secondary extension.
ML systems research	Cost-performance frontiers; constrained optimization	Compare AES architectures under quality thresholds rather than only by absolute score quality or isolated speed claims.

Table 4: Stylized CPDS illustration: lower inference cost does not necessarily yield lower effective cost.

Component	Configuration A (lower-cost inference)	Configuration B (higher-cost inference)
Inference cost per essay	\$0.01	\$0.05
Human review rate	20%	5%
Cost per human-reviewed essay	\$0.50	\$0.50
Expected review cost	\$0.10	\$0.025
CPDS	\$0.11	\$0.075

Note: Values assume the same scoring volume N , the same deployment regime, and $S(d, N) = N$ for both configurations.