Pointwise Representation Certificates for Computation and Learning

Yunbei Xu
National University of Singapore
yunbei@nus.edu.sg

Preliminary note

Abstract

This note proposes a user-friendly and still-developing language for pointwise representation certificates. It does not identify runtime, sample complexity, generalization error, and regret. Instead, it separates three objects: the resource needed to discover or verify a representation, the local size of that representation, and the residual certificate obtained after the representation is available. The residual certificate has a type: it may be computational, statistical, information-theoretic, or optimization-theoretic. Average-case and smoothed analysis put distributions on inputs; parameterized complexity fixes explicit structural parameters. Pointwise representation certificates ask a different question: for this input, trained model, optimization trajectory, or interaction, has an admissible representation been discovered that is sufficient for a smaller residual problem? The note gives self-contained definitions and explains how backdoors, tree decompositions, local nonconvex certificates, pointwise neural-network generalization, data-driven algorithm design, learning-to-optimize, statistical-query restrictions, and Bellman log-potential search fit this typed certificate view without confusing computational complexity with statistical or information complexity.

1 The separation of quantities

A task is a family

$$\mathcal{P} = \{(\mathcal{X}_n, \mathcal{Y}_n, R_n) : n \geq 1\}.$$

Here $x \in \mathcal{X}_n$ is an input or instance, $y \in \mathcal{Y}_n(x)$ is an output or certificate, and $R_n(x, y) \in \{0, 1\}$ is the verification relation. For a language L, one may take $\mathcal{Y}_n = \{0, 1\}$ and $R_n(x, y) = \mathbf{1}\{y = \mathbf{1}_L(x)\}$. For an NP search problem, $R_n(x, w) = 1$ means that w is a valid witness.

The first convention is that the word *cost* is avoided unless its unit has been specified. The relevant quantity may be time, memory, verifier calls, communication, samples, feedback, regret, excess risk, proof length, or an optimization residual. These quantities should not be added unless a theorem supplies a common scale. A pointwise certificate is therefore typed.

Definition 1.1 (Typed performance criterion)**.** *A typed performance criterion is a map* P *that assigns to an algorithm, estimator, policy, or solver a quantity with a specified unit. Examples include*

$$\mathrm{Time}(A, x), \qquad \mathrm{Risk}(\widehat{f}; \mathcal{D}), \qquad \mathrm{Reg}_T(\pi, \omega), \qquad F_x(z) - \inf_z F_x(z),$$

where the first is computational, the second statistical, the third sequential/information-theoretic, and the fourth optimization-theoretic. A theorem must state which typed criterion is being bounded.

Thus representation complexity for computation and representation complexity for learning are not the same numerical object. They share a syntax—discover a representation, charge its size, solve a residual problem—but the residual theorem is different. In computational complexity the residual may be a running time or verifier-call bound. In learning it may be an excess-risk or generalization bound. In bandits and reinforcement learning it may be a regret or information bound. The common representation language is useful precisely because it keeps these distinctions explicit.

Classical worst-case complexity studies bounds of the form

$$\inf_{A} \sup_{x \in \mathcal{X}_n} \mathrm{Time}(A, x)$$

(Roughgarden, 2021). Average-case complexity fixes a distribution $\mathcal{D}_n$ on $\mathcal{X}_n$ and studies

$$\mathbb{E}_{X \sim \mathcal{D}_n} \mathrm{Time}(A, X)$$

(Levin, 1986; Yao, 1977). This is analogous to using an external prior over inputs, but it is not a pointwise explanation of why a particular input is easy. Parameterized complexity fixes a structural parameter $\kappa(x)$ and proves

$$\mathrm{Time}(A, x) \le f(\kappa(x)) \, \mathrm{poly}(n)$$

(Downey and Fellows, 2013; Cygan et al., 2015). This is analogous to imposing a geometric or structural restriction on the instance. Pointwise representation certificates are more flexible: the useful representation may be discovered by the algorithm or interaction, but the discovery and verification resources must be charged (Xu, 2026a).

2 Admissible sufficient representations

Definition 2.1 (Admissible representation scheme)**.** *For input length n, an admissible representation scheme $\mathcal{R}$ consists of:*

(i) a representation space $\mathcal{S}_n$;

(ii) a discovery or construction procedure $G_{\mathcal{R},n}$ that maps an input or interaction history to a state $S \in \mathcal{S}_n$ and possibly a certificate ζ;

(iii) a residual interface, such as a verifier, feasible-operation map, oracle model, transition rule, objective primitive, statistical model, or Bellman update;

(iv) a residual solver or residual theorem that uses only (S, ζ) and the residual interface.

The scheme is admissible only if $\mathcal{R}$, $G_{\mathcal{R},n}$, and the residual solver or theorem are specified independently of the hidden answer or witness. If an oracle supplies solution information, that oracle must belong to the original problem and its resource usage must be charged.

Admissibility is essential. Without it, one could set S equal to the correct answer, a valid witness, or an optimal search path, and every problem would have zero residual difficulty. An analogy is PAC-Bayes prior blindness: the prior should not depend on, or already encode, the hypothesis being certified (McAllester, 1999).

Definition 2.2 (Sufficiency for a residual criterion)**.** *Let $(S, \zeta) = G_{\mathcal{R},n}(x)$.*

(a) For a decision problem, (S, ζ) is sufficient if there is a computable map $D_{\mathcal{R}}$ such that

$$\mathbf{1}_L(x) = D_{\mathcal{R}}(S, \zeta).$$

(b) For an NP search problem, it is sufficient if the exact residual witness set is determined by the state:

$$\{w : R_n(x, w) = 1\} = W_{S,\zeta}.$$

(c) For an optimization problem, it is sufficient at accuracy ε if the residual certificate proves that the specified residual solver reaches an ε-optimal point, or proves the claimed local optimum condition, using only (S, ζ).

(d) For a statistical learning problem, it is sufficient if the theorem's risk or excess-risk bound depends on the data, algorithm, and hypothesis only through (S, ζ) and stated residual quantities such as empirical error, approximation error, and optimization error.

(e) For an interactive decision problem with history H_{t-1} and state $S_t = \phi_t(H_{t-1})$, it is dynamically sufficient if the feasible operations, response law or adversarial response rule, loss or typed cost, and update close on the state:

$$\mathcal{A}_t(H_{t-1}) = \mathcal{A}_t(S_t),$$
$$P_x(\cdot \mid H_{t-1}, a) = P_{S_t}(\cdot \mid a), \qquad c_x(H_{t-1}, a) = c(S_t, a),$$
$$S_{t+1} = \tau_t(S_t, a_t, O_t).$$

The full input or full interaction history is always sufficient. A compressed state is useful only after exact closure or a certified approximation has been verified.

Definition 2.3 (Typed pointwise representation certificate)**.** *For an admissible sufficient representation scheme $\mathcal{R}$, define the typed certificate*

$$\mathsf{Cert}_{\mathcal{R}}(x) := \big(\mathrm{Disc}_{\mathcal{R}}(x), \mathrm{Size}_{\mathcal{R}}(x), \mathrm{Res}_{\mathcal{R}}(x)\big).$$

Here $\mathrm{Disc}_{\mathcal{R}}(x)$ records the resource or evidence required to discover, compute, learn, or verify the representation; $\mathrm{Size}_{\mathcal{R}}(x)$ records a description length, local entropy, rank, dimension, or other representation-size charge; and $\mathrm{Res}_{\mathcal{R}}(x)$ records the residual certificate in its own type. These entries need not share units. A theorem that turns a certificate into a scalar bound must specify an evaluation functional

$$\Psi : \ (\mathrm{Disc}, \mathrm{Size}, \mathrm{Res}) \longmapsto \textit{the typed quantity being bounded.}$$

For an admissible atlas $\mathfrak{R}$, the corresponding pointwise bound is therefore

$$\mathsf{C}_{\Psi,\mathfrak{R}}(x) := \inf_{\mathcal{R}\in\mathfrak{R}} \Psi\big(\mathsf{Cert}_{\mathcal{R}}(x)\big),$$

whenever the right-hand side is meaningful. Without such a Ψ, the tuple $\mathsf{Cert}_{\mathcal{R}}(x)$ is a structured certificate rather than a single complexity number.

A pointwise certificate is therefore not always a scalar. The components of $(\mathrm{Disc}, \mathrm{Size}, \mathrm{Res})$ may be numbers, but they may also be structured objects: an effective subspace, a local basin, a decomposition, a verifier, a Bellman potential, or a residual optimization certificate. The role of the theorem is to say which features of these objects enter the target quantity.

For example, in a running-time theorem the evaluation functional Ψ may convert discovery time, representation size, and residual dynamic-programming time into one computational resource bound. In a statistical risk theorem, Size may enter through a statistical penalty, while Disc records how the representation was found or certified and Res records empirical-risk, approximation, or optimization residuals. In pointwise deep-learning bounds, the learned effective eigenspace is part of the representation discovered by the algorithm, the associated local effective dimension contributes to Size, and the remaining empirical-risk, approximation, or optimization residual belongs to Res.

3 How pointwise certificates coexist with hardness

Pointwise certificates can explain why particular instances are easy, but they do not invalidate worst-case hardness. To avoid ambiguity, we state the claim for a polynomially checkable search relation

$$R \subseteq \{0,1\}^{\star} \times \{0,1\}^{\star},$$

with polynomially bounded witnesses and polynomial-time verifier. Its associated decision language is

$$L_R := \{x : \exists w \text{ such that } R(x,w) = 1\}.$$

Thus P and RP below refer to the decision language L_R. The class RP is the standard randomized polynomial-time class with one-sided error: no-instances are rejected with probability one, and yes-instances are accepted with probability at least a fixed positive constant, which can be amplified.

Proposition 3.1 (Pointwise tractability does not collapse worst-case hardness)**.** *Suppose R admits an admissible representation system with the following uniform polynomial guarantee. For every input x of length n, a discovery procedure finds a representation $s = \phi(x)$ in time* $\mathrm{poly}(n)$*, the representation certificate is checkable in time* $\mathrm{poly}(n)$*, and a residual deterministic solver, given (x, s), correctly decides whether a witness exists and outputs a valid witness whenever one exists, all in time* $\mathrm{poly}(n)$*. Then $L_R \in P$, and the search relation R is solvable in deterministic polynomial time.*

If instead the residual solver is randomized with one-sided error, meaning that it never outputs an invalid witness or accepts a no-instance, and on every yes-instance it outputs a valid witness with probability at least $1/2$*, with polynomial running time and standard amplification, then $L_R \in RP$.*

Proof. On input x, run the discovery procedure to obtain $s = \phi(x)$, verify the representation certificate, and then run the residual solver. In the deterministic case, the assumed soundness, completeness, and uniform polynomial running-time bounds give a polynomial-time decision algorithm for L_R and a polynomial-time search algorithm for R.

In the randomized case, reject unless the solver outputs a witness w with $R(x, w) = 1$. Soundness gives zero acceptance probability on no-instances. On yes-instances, the solver outputs a valid witness with probability at least $1/2$, and repeated independent trials amplify this success probability in polynomial time. Hence $L_R \in RP$. □

Consequently, if L_R is NP-hard under polynomial-time many-one reductions, then a uniformly polynomial deterministic certificate system of the kind in Proposition 3.1 would imply $P = NP$. A uniformly polynomial one-sided randomized certificate system would imply $NP \subseteq RP$. Thus pointwise representation certificates should not be read as worst-case algorithms for all instances unless such a collapse is intended. Their useful role is pointwise, distributional, smoothed, or trajectory-dependent: some inputs, neighborhoods, training distributions, or algorithmic paths may have small admissible certificates even though the worst-case supremum remains large.

4 Examples of typed residual certificates

4.1 Parameterized residual dynamic programs

A backdoor set B for SAT is a sufficient representation when assigning the variables in B reduces the formula to a tractable class. If $|B| = k$ and the backdoor is discovered at resource $D(x)$, brute force over the backdoor gives

$$\mathrm{Time}(x) \le D(x) + 2^k \,\mathrm{poly}(n).$$

A tree decomposition of width w gives, for many graphical and constraint problems,

$$\mathrm{Time}(x) \le D(x) + n \exp(O(w)).$$

These are standard parameterized-algorithmic examples of admissible sufficient representations: the representation is useful because it is checkable, discoverable or supplied with its discovery resource charged, and sufficient for the residual dynamic program (Downey and Fellows, 2013; Cygan et al., 2015; Roughgarden, 2021).

4.2 Local nonconvex optimization certificates

For nonconvex optimization, let $F_x : \mathbb{R}^m \to \mathbb{R}$ be the objective. A sufficient local optimization state may contain a region U_S, an initialization $z_0(S)$, and constants (L_S, μ_S) such that the algorithm stays in U_S, F_x has L_S-Lipschitz gradient on U_S, and the Polyak–Lojasiewicz inequality holds on U_S:

$$\frac{1}{2}\|\nabla F_x(z)\|^2 \ge \mu_S\big(F_x(z) - F_x^\star(U_S)\big), \qquad z \in U_S.$$

Then gradient descent with step size $1/L_S$ reaches error ε in

$$O\left(\frac{L_S}{\mu_S} \log \frac{F_x(z_0) - F_x^\star(U_S)}{\varepsilon}\right)$$

iterations. This bypasses global nonconvex hardness only pointwise: the certificate proves that this instance and initialization lie in a tractable invariant basin. It does not claim that all basins can be found or that the full landscape is easy.

4.3 Statistical pointwise representations and deep learning

In supervised learning the target criterion is usually population risk, not runtime. A pointwise statistical certificate should therefore be written as an oracle inequality, not as a computational time bound. A typical form is

$$\begin{aligned}\mathrm{Risk}(\widehat{f}; \mathcal{D}) - \mathrm{Risk}(f^\star; \mathcal{D}) \le\ & \mathrm{Approx}(S) + \mathrm{AlgErr}(\widehat{f}; S) \\ & + \mathrm{Stat}(\mathrm{Size}(S), n, \delta),\end{aligned}$$

with probability at least $1-\delta$. Here S is the learned or selected representation, $\mathrm{Approx}(S)$ is the bias or approximation price of using that representation, $\mathrm{AlgErr}(\widehat{f}; S)$ is the empirical or optimization residual such as an ERM gap, and $\mathrm{Stat}(\mathrm{Size}(S), n, \delta)$ is the statistical penalty. The discovery resource used to obtain S, for example training time, is a separate computational quantity unless the theorem explicitly combines it with risk.

Recent pointwise generalization theory for deep neural networks fits this template. The effective eigenspace or spectrum of the learned feature representation is a candidate state S; its local effective dimension is a representation-size term; and the remaining residual is the pointwise generalization error within the represented class, together with the corresponding approximation and optimization errors (Li and Xu, 2026). The final theorem is a statistical generalization certificate. It becomes a computational certificate only after one also bounds the resource needed to find or verify the effective eigenspace and to control the empirical or optimization residual. Thus the representation language is shared, but the typed conclusion is different: feature-spectrum complexity controls a risk bound, not the running time of training unless training resources are separately charged.

4.4 Bellman log-potential search and sequential decisions

For planning, proof search, program synthesis, reasoning, bandits, or reinforcement learning, let Y_x be the correct answer, branch, proof, plan, policy, or certificate. Suppose a sufficient search state S_t maintains an admissible distribution q_t over candidate indices. A log-potential certificate uses

$$V_t(S_t) + \gamma \log \frac{1}{q_t(Y_x)}.$$

If each operation satisfies the verified Bellman inequality

$$c_t(S_t, A_t) + \mathbb{E} V_{t+1}(S_{t+1}) - V_t(S_t) \le \gamma \, \mathbb{E} \log \frac{q_{t+1}(Y_x)}{q_t(Y_x)} + \varepsilon_t,$$

then telescoping gives

$$\mathbb{E} \sum_{t=1}^{T} c_t \le V_1(S_1) + \gamma \log \frac{1}{q_1(Y_x)} + \sum_{t=1}^{T} \varepsilon_t,$$

up to a nonnegative terminal potential. The residual criterion here may be a search resource, a regret bound, or an information bound, depending on the problem. This is the log-potential form behind PAC–Bayes, exponential weights, and Bellman-sufficient information complexity for sequential decisions (Xu, 2026b). It is valid only when the state update, distribution update, and Bellman inequality are part of the certificate. The typed conclusion is a search-resource, regret, or information bound according to the problem statement; the computational cost of solving the Bellman program is separate unless it is also charged.

5 Existing theories through the typed lens

5.1 Average-case and smoothed analysis

Average-case complexity studies $\mathbb{E}_{X \sim \mathcal{D}_n} \operatorname{Time}(A, X)$. A pointwise representation theorem instead proves a bound on a typed certificate $\mathsf{Cert}_{\mathcal{R}}(X)$ and then studies its distribution. Smoothed analysis studies $X = \bar{x} + \xi$, with adversarial center $\bar{x}$ and random perturbation ξ (Spielman and Teng, 2004). A pointwise representation statement has the form

$$\sup_{\bar{x}} \mathbb{P}_{\xi} \{ \mathsf{Cert}_{\mathcal{R}}(\bar{x} + \xi) \notin \mathcal{G}_u \} \le \delta(u),$$

where $\mathcal{G}_u$ is the event that the discovery resource, representation size, and residual resource are within the claimed range. Perturbation does not remove worst-case hardness; it makes bad certificates unlikely under the perturbation model.

5.2 Data-driven algorithm design

Data-driven algorithm design chooses an algorithmic parameter from training instances and proves that its performance generalizes to future instances (Balcan, 2020; Balcan et al., 2021; Gupta and Roughgarden, 2020; Haussler, 1992). If Θ is a finite parameter class and losses lie in $[0,1]$, then with m training instances,

$$\sup_{\theta\in\Theta}\left|\mathbb{E}\,\mathrm{Cost}(A_\theta, Z) - \frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}(A_\theta, Z_i)\right| \lesssim \sqrt{\frac{\log|\Theta| + \log(1/\delta)}{m}}$$

with high probability. This is a statistical generalization theorem about selecting an algorithmic parameter. It does not by itself describe the pointwise representation used by the selected algorithm on a future instance. A pointwise question can be asked after selection: which sufficient state does $A_{\hat\theta}$ discover on the new instance, what resource is required, and what residual problem remains?

5.3 Learning-to-optimize

Learning-to-optimize trains an update rule or iterative policy for a family of optimization problems (Chen et al., 2022; Sucker and Ochs, 2025; Sambharya and Stellato, 2025). The learned object is a computational policy, not just a static algorithmic parameter. A typed certificate has two layers. First, a statistical theorem controls how the learned optimizer generalizes across tasks. Second, a pointwise residual certificate controls convergence on a new task, for example by a local PL basin, a descent certificate, or a Bellman log-potential inequality. Deployment time, number of iterations, and memory are computational resources; task-generalization is a statistical quantity. They can both be studied, but they should not be merged without an explicit theorem.

5.4 Statistical-query restrictions

Statistical-query algorithms access expectations through noisy query oracles (Kearns, 1998; Feldman, 2017). This is a formal admissibility restriction: the state may contain only quantities estimable by the SQ oracle to the required tolerance. A high SQ dimension can be read as a lower bound on the existence of small SQ-accessible sufficient representations, even when sample-theoretic complexity alone is small. This is one of the cleanest links between statistical and computational complexity, because the statistical query interface is also a computational restriction.

6 Conclusion

Pointwise representation certificates are not a single replacement for computational complexity or statistical learning theory. It is a typed certificate language. Average-case analysis weights inputs; parameterized complexity restricts inputs by explicit structural parameters; pointwise representation certificates ask what useful state can be discovered on this input, trained model, or interaction, charge that discovery, and prove that the residual problem closes on the state. The framework separates computation, representation size, and statistical information. Computation discovers or exploits a representation; the representation has local size; and the residual certificate may be computational, statistical, optimization-theoretic, or information-theoretic. This provides a user-friendly and still-developing language for explaining the tractability of particular hard-looking instances, nonconvex objectives, combinatorial structures, learned optimizer states, neural representations, and reasoning searches, while remaining consistent with worst-case hardness.

References

Maria-Florina Balcan. Data-driven algorithm design. In *Beyond the Worst-Case Analysis of Algorithms*, Cambridge University Press, 2020. See also arXiv:2011.07177.

Maria-Florina Balcan, Dan DeBlasio, Travis Dick, Carl Kingsford, Tuomas Sandholm, and Ellen Vitercik. How much data is sufficient to learn high-performing algorithms? Generalization guarantees for data-driven algorithm design. In *STOC*, 2021.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.

Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

Vitaly Feldman. A general characterization of the statistical query complexity. In *COLT*, 2017.

Rishi Gupta and Tim Roughgarden. Data-driven algorithm design. *Communications of the ACM*, 63(6):87–94, 2020.

David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.

Leonid A. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

Shaojie Li and Yunbei Xu. Pointwise generalization in deep neural networks. arXiv:2605.18598, 2026.

David A. McAllester. PAC-Bayesian model averaging. In *COLT*, 1999.

Tim Roughgarden, editor. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021.

Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

Rajiv Sambharya and Bartolomeo Stellato. Data-driven performance guarantees for classical and learned optimizers. *Journal of Machine Learning Research*, 26(171):1–49, 2025.

Michael Sucker and Peter Ochs. Learning-to-optimize with PAC-Bayesian guarantees. *Journal of Machine Learning Research*, 26, 2025.

Yunbei Xu. Pointwise complexity for Gaussian fields: Upper envelopes, algorithmic lower bounds, and separation. arXiv:2606.07931, 2026.

Yunbei Xu. Bellman-sufficient information complexity. arXiv:2606.11171, 2026.

Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. In *FOCS*, 1977.