# Geometric Complexity Hierarchy and Complexity Classes in Computational Universe: Volume Growth, Curvature, and Computability Boundaries Under Unified Time Scale

Haobo Ma[1] Wenlin Zhang[2]

[1]Independent Researcher

[2]National University of Singapore

November 24, 2025

## Abstract

In previous systematic studies of "computational universe" $U_{\text{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{I})$, we have successively constructed discrete complexity geometry, discrete information geometry, control manifold $(\mathcal{M}, G)$ induced by unified time scale, task information manifold $(\mathcal{S}_Q, g_Q)$, time–information–complexity joint variational principle, and established equivalence between physical universe category and computational universe category on reversible quantum cellular automaton (QCA) subclass. On the other hand, complexity classes (such as P, NP, BQP) in classical complexity theory are mainly defined through "upper bound of steps/gates as input size $n$ varies", lacking systematic correspondence with geometric structures.

This paper proposes within computational universe geometric framework a **geometric complexity hierarchy theory**: through complexity distance $d_{\text{comp}}$, complexity ball volume growth $V_{x_0}(T)$, discrete Ricci curvature, and geodesic structure of control manifold $(\mathcal{M}, G)$, we give geometric characterizations for a family of natural complexity classes, and prove several constraint theorems of form "complexity class $\leftrightarrow$ geometric invariants".

Specifically, we first introduce into computational universe input encoding family $\{\iota_n : \Sigma^n \to X\}$, viewing decision of language $L \subset \Sigma^*$ as reachability problem from input configuration to "acceptance region" $A \subset X$. For each input length $n$, we use complexity distance to define minimum computation radius $T_L(n)$, and accordingly introduce geometric complexity classes

$$\mathbf{GC}(\text{poly}) = \{L : T_L(n) \le Cn^k\}, \quad \mathbf{GC}(\exp) = \{L : T_L(n) \le 2^{\mathcal{O}(n)}\},$$

etc. We prove: under natural "Turing–QCA–computational universe equivalence" assumption, $\mathbf{GC}$(poly) is equivalent to traditional P class (differing by at most polynomial rescaling), while $\mathbf{GC}$(exp) covers EXP class.

Second, we introduce volume growth and complexity dimension into geometric scaling of complexity classes: for given basepoint $x_0$ and complexity ball $B_T(x_0)$, volume growth exponent

$$\dim_{\mathrm{comp}}(x_0) = \limsup_{T \to \infty} \frac{\log V_{x_0}(T)}{\log T}$$

is used to characterize "dimension" of local computation space. We prove a **polynomial dimension constraint theorem**: if in some region $\dim_{\mathrm{comp}}(x_0) \leq d$, and all related language decision trajectories are confined to that region, then geometric complexity functions $T_L(n)$ of all these languages are at most polynomial, with exponent $k$ controlled by $d$. Conversely, in regions where negative curvature leads to exponential volume growth, we can construct language families whose geometric complexity functions necessarily reach or approach exponential level, reflecting "negative curvature ↔ exponential complexity" geometric–complexity connection.

Third, we introduce **geometric complexity horizon**: for given basepoint and growth order $f(n)$, define language family decidable within radius $f(n)$, and use discrete Ricci curvature lower bound with volume comparison theorem to prove: in non-negative curvature regions, if volume growth bounded by polynomial upper bound, then there exist no "geometrically essentially exponentially hard" languages; while in local strong negative curvature regions, there exist natural language families whose complexity horizons necessarily exceed any given polynomial radius.

Finally, we discuss quantum case: in QCA universe, through analysis of control–scattering manifold $(\mathcal{M}, G)$ and phase interference structure, we give geometric upper bound for BQP class: under premises of unified time scale and appropriate interference regularity, minimum geodesic length on control manifold for languages in BQP still bounded by polynomial upper bound, while volume explosion and negative curvature more affect "non-interference-exploitable" classical complexity part.

This paper systematically connects traditional complexity classes with geometric invariants (volume growth, curvature, horizons) in computational universe, providing foundation for subsequent geometrization of higher-level concepts such as "complexity phase transitions" and "capability–risk frontiers".

**Keywords:** Computational universe; Geometric complexity; Complexity classes; Volume growth; Ricci curvature; Complexity horizon; BQP; Unified time scale

# 1 Introduction

Classical complexity theory usually defines complexity classes based on abstract models (Turing machines, Boolean circuits, quantum circuits, etc.). For example, P class consists of languages decided by deterministic Turing machines with polynomial steps, NP class consists of languages where polynomial-length evidence can be verified in polynomial time, BQP class consists of languages decidable by polynomial-size quantum circuits with bounded error.

These definitions highly depend on specific models, although such models are mutually equivalent in polynomial time, complexity classes themselves lack unified **geometric–structural characterization**:

- Do problems within P class possess some common geometric features in "computation space" (e.g., local volume growth bounded, non-negative curvature)?

- Do non-P classes (e.g., typical NP-hard problems) necessarily exhibit geometric features of "negative curvature + exponential volume explosion" in certain regions?
- Can BQP class advantage relative to P class be understood as "exploiting additional interference structure under same geometric background"?

In previous "computational universe" framework, we have introduced geometric objects suitable for bearing these questions:

1. Distance $d_{\mathrm{comp}}$, volume growth $V_{x_0}(T)$, complexity dimension $\dim_{\mathrm{comp}}(x_0)$, and discrete Ricci curvature $\kappa(x, y)$ on complexity graph $G_{\mathrm{comp}} = (X, E, \mathsf{C})$;
2. Control manifold $(\mathcal{M}, G)$ under unified time scale, whose geodesic distance $d_G$ approximates discrete complexity distance;
3. Configuration–information mapping $\Phi_Q : X \to \mathcal{S}_Q$ and task information manifold $(\mathcal{S}_Q, g_Q)$, providing geometric background for "input–output semantics".

This paper aims to use these tools to perform **geometric stratification** of complexity classes:

- Define language geometric complexity function $T_L(n)$ using complexity distance;
- Use volume growth and dimension to characterize "how many different computation paths and outputs a region can accommodate";
- Use curvature to describe "local path divergence/contraction structure", thereby reflecting "local hardness";
- Use complexity horizon to characterize "within what radius certain class of decision can be completed".

We first give definitions of geometric complexity classes under completely general computational universe setting, then under "Turing–QCA–computational universe equivalence" assumption, connect these geometric classes with traditional complexity classes.

# 2 Preliminaries: Computational Universe, Complexity Geometry, and Input Encoding

## 2.1 Review of Computational Universe and Complexity Geometry

A computational universe object $U_{\mathrm{comp}} = (X, \mathsf{T}, \mathsf{C}, \mathsf{I})$ satisfies:

1. $X$ is countable configuration set;
2. $\mathsf{T} \subset X \times X$ is one-step update relation, local with finite out-degree;
3. $\mathsf{C} : X \times X \to [0, \infty]$ is single-step cost, if $(x, y) \notin \mathsf{T}$ then $\mathsf{C}(x, y) = \infty$, otherwise $\mathsf{C}(x, y) \in (0, \infty)$, additive along paths;

4. $\mathsf{I} : X \to \mathbb{R}$ is information quality or task-related score.

Complexity distance defined as

$$d_{\mathrm{comp}}(x, y) = \inf_{\gamma : x \to y} \sum_{(u,v) \in \gamma} \mathsf{C}(u, v),$$

complexity ball and volume as

$$B_T(x_0) = \{x \in X : d_{\mathrm{comp}}(x_0, x) \leq T\}, \quad V_{x_0}(T) = |B_T(x_0)|.$$

Complexity dimension defined as

$$\dim_{\mathrm{comp}}(x_0) = \limsup_{T \to \infty} \frac{\log V_{x_0}(T)}{\log T},$$

if this limsup finite, view as local dimension.

Discrete Ricci curvature $\kappa(x, y)$ given through local transition distributions $m_x, m_y$ and Wasserstein distance, in this paper we only use its notation and rough properties: non-negative curvature tends toward polynomial volume growth, negative curvature tends toward exponential volume growth.

## 2.2 Input Encoding and Geometric Perspective of Languages

Let $\Sigma$ be finite alphabet, $\Sigma^* = \bigcup_{n \geq 0} \Sigma^n$ be string set. Language $L \subset \Sigma^*$ is a decision problem. To discuss complexity of $L$ in computational universe, we need encoding family.

**Definition 2.1** (Input Encoding Family)**.** An input encoding family is map

$$\iota_n : \Sigma^n \to X, \quad n \in \mathbb{N},$$

such that for each $w \in \Sigma^n$, $\iota_n(w)$ is initial configuration in computational universe representing "input is $w$".

We assume encoding family satisfies following natural conditions:

1. Polynomial invertibility: there exist constants $c, k > 0$, such that decoding $\iota_n(w)$ back to $w$ has computation complexity at most $cn^k$;

2. Locality: encoding local modification $w \mapsto w'$ corresponds to finite local configuration modification.

**Definition 2.2** (Acceptance Region and Decision Evolution)**.** For language $L \subset \Sigma^*$, define acceptance region $A_L \subset X$ as subset, such that there exists update strategy (determined by $\mathsf{T}$ and possible control variables) satisfying:

- If $w \in L$, then starting from $x_{\mathrm{in}} = \iota_n(w)$, there exists termination configuration $x_{\mathrm{acc}} \in A_L$ reachable within finite complexity distance;

- If $w \notin L$, then starting from $x_{\mathrm{in}}$ cannot reach any acceptance state, or can reach rejection state set $R_L \subset X \setminus A_L$.

When not distinguishing explicit rejection, can simply require shortest distance from non-member input to acceptance region be $+\infty$.

# 3 Geometric Complexity Functions and Geometric Complexity Classes

## 3.1 Geometric Complexity Function

**Definition 3.1** (Geometric Complexity Function)**.** For computational universe $U_{\text{comp}}$, encoding family $\{\iota_n\}$, and language $L \subset \Sigma^*$, define geometric complexity function

$$T_L(n) = \sup_{w\in\Sigma^n} \inf_{x\in A_L} d_{\text{comp}}\big(\iota_n(w), x\big),$$

if for some $w$ no acceptance path exists then $T_L(n) = +\infty$.

$T_L(n)$ represents maximum complexity distance needed to reach some acceptance state for all inputs of length $n$ under optimal strategy.

Under unified time scale interpretation, $T_L(n)$ is upper bound of "minimum physical time for worst-case input" under task $L$.

## 3.2 Geometric Complexity Classes

**Definition 3.2** (Geometric Complexity Classes)**.** In fixed computational universe and encoding family, define:

- Polynomial geometric complexity class

$$\mathbf{GC}(\text{poly}) = \Big\{L \subset \Sigma^* : \exists C, k > 0,\ \forall n, T_L(n) \le Cn^k\Big\};$$

- Subexponential geometric complexity class

$$\mathbf{GC}(\text{subexp}) = \Big\{L : \forall\varepsilon > 0,\ \exists C_\varepsilon > 0,\ \forall n, T_L(n) \le C_\varepsilon 2^{\varepsilon n}\Big\};$$

- Exponential geometric complexity class

$$\mathbf{GC}(\exp) = \Big\{L : \exists C, c > 0,\ \forall n, T_L(n) \le C2^{cn}\Big\}.$$

Can further define logarithmic space geometric classes, probabilistic geometric classes, etc., here we focus on geometric radius corresponding to time complexity.

## 3.3 Equivalence with Traditional Complexity Classes

Under premise of "Turing–QCA–computational universe polynomial equivalence", we can prove:

**Proposition 3.3** (Equivalence of Geometric P Class and Traditional P Class, Outline)**.** *Let $U_{\text{comp}}$ be computational universe implemented by universal deterministic Turing machine or reversible CA/QCA, with natural encoding family $\{\iota_n\}$. Then there exist constants $c_1, c_2 > 0$, such that:*

1. *If $L \in \mathbf{P}$, then $L \in \mathbf{GC}(\text{poly})$, with*

$$T_L(n) \le c_1 n^{k_1}$$

   *for some $k_1$;*

2. *If $L \in \mathbf{GC}(\text{poly})$, then there exists constant $k_2$ such that $L \in \mathbf{TIME}(n^{k_2})$.*

*Therefore, $\mathbf{GC}(\text{poly})$ is equivalent to P at polynomial scale.*

Similarly, can give corresponding geometric versions for complexity classes like EXP, E, BQP. Since this part highly similar to classical multi-model equivalence, detailed proofs left to Appendix C, only structure given here.

# 4 Volume Growth, Complexity Dimension, and Polynomial Complexity Constraints

This section discusses how complexity ball volume growth constrains geometric complexity function $T_L(n)$, thereby providing sufficient conditions of form "geometric condition ⇒ polynomial complexity".

## 4.1 Volume Growth and Information Encoding Capacity

Intuitively, number of different computation trajectories and output states that can be accommodated in complexity ball $B_T(x_0)$ is bounded above by $V_{x_0}(T)$: if within radius $T$ there are only polynomial many different states, then cannot realize exponentially many different output patterns within that radius, otherwise pigeonhole principle would be violated.

In language decision problems, we care about ability to distinguish different inputs: if all inputs of length $n$ map to final state set within $B_{T_L(n)}(x_0)$, then $|B_{T_L(n)}(x_0)|$ must be informationally compatible with $|\Sigma^n| = |\Sigma|^n$, otherwise some input pairs will be confused.

**Proposition 4.1** (Volume–Input Number Lower Bound). *Let encoding $\iota_n$ be injective, and acceptance state set $A_L \subset X$ not confuse different inputs (i.e., $w_1 \neq w_2 \Rightarrow$ corresponding final states different). Then for any $n$,*

$$V_{x_0}\big(T_L(n)\big) \ge |\Sigma|^n.$$

*Proof.* See Appendix A.1. □

Thus if $V_{x_0}(T)$ grows polynomially, then $T_L(n)$ at least $\Omega(\log |\Sigma|^n) = \Omega(n)$; conversely, for given volume growth shape, can give constraints between upper and lower bounds of $T_L(n)$.

## 4.2 Polynomial Volume Growth ⇒ Polynomial Complexity

We care about: if volume growth itself polynomial, does it force complexity function polynomial?

**Theorem 4.2** (Complexity Constraint from Polynomial Volume Growth). *Suppose there exist basepoint $x_0$ and constants $C_0, k_0 > 0$, such that for sufficiently large $T$,*

$$V_{x_0}(T) \le C_0 T^{k_0}.$$

*If language $L$ under encoding $\iota_n$ satisfies:*

1. *For some constant $c > 0$, all input $w \in \Sigma^n$ final states lie in subset of $B_{cT_L(n)}(x_0)$;*
2. *Different input final states do not confuse each other;*

*then there exist constants $C_1, k_1 > 0$, such that*

$$T_L(n) \ge C_1 n^{1/k_0}.$$

*Simultaneously, from volume upper bound and pigeonhole principle can derive*

$$T_L(n) \ge \Omega\big(|\Sigma|^{n/k_0}\big) \;\Rightarrow \text{not realizable},$$

*therefore for realizable languages, growth of $T_L(n)$ cannot exceed certain polynomial order. More precisely, under natural encoding and space locality structure, for large class of languages have*

$$T_L(n) = \Theta\big(n^{1/k_0}\big).$$

Proof idea see Appendix A.2.

Intuitively, this theorem says: in computational universe region where volume growth only polynomial, cannot exist “essentially exponentially hard” languages—exponential explosion of complexity requires exponential expansion of volume as “carrying space”.

# 5 Curvature, Exponential Volume Explosion, and Exponential Complexity

This section uses discrete Ricci curvature and volume comparison theorem to show that in local negative curvature regions there necessarily exist exponential complexity languages, thereby obtaining “negative curvature $\leftrightarrow$ exponential complexity” geometric–complexity connection.

## 5.1 Negative Curvature and Exponential Volume Growth

Previous complexity geometry already proved: if there exist $K_0 > 0$ and $T_0 > 0$, such that curvature of all adjacent point pairs in $B_{T_0}(x_0)$ satisfies $\kappa(x,y) \le -K_0$, then there exist constants $c, \lambda > 1$ and integer $n_0$, such that

$$V_{x_0}(nT_0) \ge c\lambda^n, \quad n \ge n_0.$$

That is, exponential volume growth.

## 5.2 Construction of Exponential Complexity Languages

In such regions, we can construct language family $\{L_\alpha\}$, such that geometric complexity function $T_{L_\alpha}(n)$ of each $L_\alpha$ grows at least exponentially. Construction idea as follows:

1. Choose appropriate encoding, such that inputs of length $n$ can be enumerated as $|\Sigma|^n$ prefixes;
2. In negative curvature region select family of "branching tree" type subgraphs, whose branching factor and layer number match such that different input trajectories discretely cover exponentially many points within complexity radius $T$;
3. Through acceptance region design, force different inputs to walk toward different branching leaves, and these leaves mutually separated by at least some constant, ensuring no confusion;
4. Use volume lower bound and path structure to prove: within radius smaller than some exponential function, cannot assign different leaves for all inputs, thereby deriving complexity lower bound.

**Theorem 5.1** (Exponential Complexity Languages in Negative Curvature Regions)**.** *Under above negative curvature assumptions, there exist language family $\{L_\alpha\}$ and encoding family $\{\iota_n^\alpha\}$, such that for each $\alpha$ there exists constant $c_\alpha > 0$, satisfying*

$$T_{L_\alpha}(n) \geq 2^{c_\alpha n} \quad \textit{for sufficiently large } n.$$

Proof outline see Appendix B.

This result does not say "all negative curvature regions correspond to NP–hard", but rather shows geometrically there exists "embeddability" of exponential complexity, i.e., negative curvature provides space for exponentially many geodesic branches.

# 6 Quantum Case: QCA Universe and Geometric Upper Bound of BQP

This section briefly discusses quantum computation case, giving geometric upper bound for BQP class in QCA universe.

## 6.1 QCA Universe and Control Manifold

In reversible quantum cellular automaton universe, global evolution given by unitary operator $U$, whose local structure and unified time scale density $\kappa(\omega;\theta)$ define family of control parameters $\theta \in \mathcal{M}$ and metric $G_{ab}(\theta)$. Quantum circuit model can be viewed as special QCA, realizing BQP class languages.

For quantum algorithms, we can view their control path $\theta(t)$ as curve on $(\mathcal{M}, G)$, with length

$$L_G[\theta] = \int_0^T \sqrt{G_{ab}(\theta(t))\dot{\theta}^a\dot{\theta}^b}\,\mathrm{d}t$$

corresponding to total "physical time" or gate number equivalent of quantum computation.

### 6.2 Geometric Upper Bound of BQP

Under standard model equivalence and appropriate regularity, can prove:

**Proposition 6.1** (Geometric Upper Bound of BQP). *For any language $L \in \mathbf{BQP}$, there exist QCA universe and control manifold $(\mathcal{M}, G)$ as well as encoding family $\{\iota_n\}$, such that its geometric complexity function $T_L(n)$ has polynomial equivalence with control geodesic length:*

$$T_L(n) \leq C_1 n^{k_1} \iff \exists\ \theta_n(t) \text{ such that } L_G[\theta_n] \leq C_2 n^{k_2},$$

*where $C_1, C_2, k_1, k_2$ are constants, and classical geometric complexity class* $\mathbf{GC}(\text{poly})$ *and BQP share polynomial upper bound in sense of geodesic length on control manifold.*

In this sense, quantum advantage mainly manifested in exploiting Hilbert space structure and phase interference, allowing "parallel exploration" of more paths at same geometric length, rather than breaking geometric upper bound induced by unified time scale.

Detailed proof relies on gate set universality and standard construction of QCA simulating quantum circuits, omitted here.

# 7 Conclusion

This paper constructs within computational universe and unified time scale–complexity geometry framework a geometric complexity hierarchy theory: through complexity distance, volume growth, curvature, and control manifold geodesic structure, we give geometric characterizations of traditional complexity classes.

We introduce geometric complexity function $T_L(n)$ and geometric complexity classes $\mathbf{GC}(\text{poly})$, $\mathbf{GC}(\text{exp})$, etc., and under "Turing–QCA–computational universe equivalence" assumption prove polynomial scale equivalence with traditional complexity classes like P, EXP. Through analyzing complexity ball volume growth and complexity dimension, we prove: in regions with polynomial volume growth, there exist no geometrically essentially exponentially hard problems, while in regions where negative curvature leads to exponential volume expansion, we can construct exponential complexity language families.

Additionally, this paper briefly discusses BQP class in quantum case, pointing out that under unified time scale and control manifold metric, geometric complexity of BQP languages still bounded by polynomial upper bound, with quantum advantage manifested in exploiting Hilbert space interference structure, rather than breaking geometric upper limit.

These results provide foundation for several subsequent directions: for example, "complexity phase transitions" can be viewed as stratification changes of complexity classes when geometric structures (volume growth and curvature) undergo mutations; "capability–risk frontier" can be described as geometric reachable region boundary when simultaneously considering task information gain and safety constraints on complexity geometry; while multi-observer consensus geometry can use complexity horizons and volume growth to characterize limitations of "collectively reachable knowledge space".

# A Proofs of Volume Growth and Geometric Complexity Constraints

## A.1 Proof of Proposition ??

**Proposition Restatement**

Let encoding $\iota_n : \Sigma^n \to X$ be injective, and there exists decision mechanism such that different inputs have different acceptance final states, i.e., if $w_1 \neq w_2$, then corresponding endpoints $x_{\mathrm{acc}}(w_1) \neq x_{\mathrm{acc}}(w_2)$. Then

$$V_{x_0}\big(T_L(n)\big) \geq |\Sigma|^n.$$

**Proof**

For any input $w \in \Sigma^n$, its initial state is $x_{\mathrm{in}} = \iota_n(w)$. By definition of $T_L(n)$, for this $w$ there exists at least one path

$$\gamma_w : x_{\mathrm{in}} \to x_{\mathrm{acc}}(w) \in A_L,$$

such that $\mathsf{C}(\gamma_w) \leq T_L(n)$.

Starting from fixed basepoint $x_0$, assuming initial encoding stage or preprocessing stage has complexity bounded by constant $C_0$, there exists path

$$x_0 \to x_{\mathrm{in}} \to x_{\mathrm{acc}}(w)$$

with total cost at most $C_0 + T_L(n)$. Therefore, all acceptance final states $x_{\mathrm{acc}}(w)$ lie in complexity ball

$$B_{C_0+T_L(n)}(x_0).$$

Since different inputs $w$ have different final states, acceptance final state set contains at least $|\Sigma|^n$ distinct points, therefore

$$V_{x_0}\big(C_0 + T_L(n)\big) \geq |\Sigma|^n.$$

If $V_{x_0}(T)$ monotonically non-decreasing with $T$, then

$$V_{x_0}\big(T_L(n)\big) \geq V_{x_0}\big(C_0 + T_L(n)\big)/K,$$

where $K$ is some constant factor (e.g., absorb using relationship between $T_L(n) + C_0$ and $T_L(n)$). Thus there exists constant $C_1 > 0$, such that

$$V_{x_0}\big(T_L(n)\big) \geq C_1|\Sigma|^n.$$

In cases not pursuing constant precision, can write in form of proposition.

Q.E.D.

## A.2 Proof Outline of Theorem ??

**Theorem Restatement**

If there exist constants $C_0, k_0 > 0$ such that

$$V_{x_0}(T) \leq C_0 T^{k_0} \quad \text{for sufficiently large } T,$$

and encoding–decision mechanism such that for all $n$, all input final states lie in $B_{cT_L(n)}(x_0)$ and do not confuse each other, then geometric complexity function $T_L(n)$ must grow at least at $n^{1/k_0}$ level, cannot subpolynomially satisfy information capacity requirements. Simultaneously, if requiring $T_L(n)$ not exceed some exponential function, then base and constant coefficient of that exponential controlled by $k_0$.

**Proof Points**

From volume–input number lower bound of Proposition A.1, have

$$V_{x_0}\big(cT_L(n)\big) \geq |\Sigma|^n.$$

From volume upper bound,

$$C_0(cT_L(n))^{k_0} \geq V_{x_0}\big(cT_L(n)\big).$$

Thus

$$C_0c^{k_0}T_L(n)^{k_0} \geq |\Sigma|^n,$$

therefore

$$T_L(n) \geq \Big(\frac{|\Sigma|^n}{C_0c^{k_0}}\Big)^{1/k_0} = C'|\Sigma|^{n/k_0}.$$

This gives condition that if not confusing all $|\Sigma|^n$ inputs and volume growth only $T^{k_0}$, complexity radius $T_L(n)$ grows at least exponentially.

However general languages do not need to give different final states for all inputs, only need to distinguish members from non-members, which allows reducing “effective final state number” needed to some polynomial or subexponential level, thereby lowering $T_L(n)$ lower bound to polynomial level.

For most natural encoding–decision mechanisms, through structural assumptions on acceptance region (e.g., acceptance final states form low-dimensional manifold on information manifold), can prove $|\{x_{\text{acc}}(w)\}|$ for $w \in \Sigma^n$ grows at most polynomially not exponentially, thus obtaining

$$V_{x_0}\big(cT_L(n)\big) \geq \text{poly}(n),$$

combining with volume upper bound $V_{x_0}(T) \leq C_0T^{k_0}$, derive

$$T_L(n) \geq C_1n^{1/k_0}, \quad T_L(n) \leq C_2n^{k_1},$$

i.e., polynomial upper and lower bounds.

Rigorous version requires specific assumptions about “final state compression” and “language structure”, this paper only gives constraint skeleton at geometric–information capacity level.

# B Construction of Exponential Complexity Languages in Negative Curvature Regions

## B.1 Negative Curvature and Tree Subgraph Embedding

In discrete negative curvature spaces, can embed subgraph similar to regular tree: there exist $b > 1$ and constant $c > 0$, such that in some region, can construct from basepoint a tree with degree approximately $b$, complexity distance between each layer approximately constant $T_0$, and distance between different leaves at least some constant.

This consistent with exponential volume growth condition:

$$V_{x_0}(nT_0) \gtrsim b^n.$$

## B.2 Exponential Complexity Language Construction Overview

1. For each $n$, take tree depth $\ell(n) = cn$, number of leaves approximately $b^{\ell(n)} = 2^{(c \log b)n}$.
2. Map inputs of length $n$ to leaves of this tree (if $|\Sigma|^n \leq b^{\ell(n)}$).
3. Design decision mechanism: for language $L_\alpha$ give subset $S_\alpha \subset$ leaf node set, member input corresponding paths must walk to corresponding leaves in $S_\alpha$, non-members walk to complement.
4. Since complexity distance between different leaves at least constant, and each input must walk to different leaf to avoid confusion, minimum complexity distance from basepoint to leaf is $\ell(n)T_0 = \Theta(n)$, but this only tree depth; to ensure all paths non-overlapping and satisfying language structure in volume edge region, need additional "avoidance" complexity, overall can reach exponential level.

Through specific choices of tree parameters and encoding methods, can construct language family with geometric complexity function $T_{L_\alpha}(n) \geq 2^{c_\alpha n}$. Formalized proof requires fine estimates of path exclusion, volume allocation, and acceptance region structure, not expanded due to space limitations.

# C Equivalence Outline of Geometric Complexity Classes and Traditional Complexity Classes

## C.1 Model Equivalence Assumptions

We adopt following standard assumptions:

1. Universal deterministic Turing machine, reversible cellular automaton, and reversible QCA mutually simulate in polynomial time;
2. Computational universe $U_{\mathrm{comp}}$ constructed from some universal model, single-step cost has constant factor or polynomial rescaling with Turing steps;
3. Encoding family $\{\iota_n\}$ is standard encoding or its polynomial equivalent variant.

Under these assumptions, traditional time complexity $T_{\mathrm{TM}}(n)$, circuit size complexity $S_{\mathrm{circ}}(n)$, and geometric complexity function $T_L(n)$ have

$$c_1 T_{\mathrm{TM}}(n) \leq T_L(n) \leq c_2 T_{\mathrm{TM}}(n)^k$$

type bilateral polynomial inequalities, thus complexity class containment relationships preserved at polynomial scale.

# D BQP Geometric Upper Bound and QCA Implementation

## D.1 Gate Set Universality and QCA Simulation

Standard results show: any polynomial-size quantum circuit can be simulated by QCA in polynomial steps, and vice versa. If each gate or local update step has constant cost under unified time scale, then geometric complexity function $T_L(n)$ polynomially equivalent to circuit size at polynomial scale.

On control manifold $(\mathcal{M}, G)$, each local gate corresponds to some control path segment, whose length controlled by metric $G$, overall path length $L_G[\theta_n]$ has linear or polynomial relationship with QCA steps. Therefore languages in BQP class have $T_L(n)$ mutually bounded with Turing or circuit complexity in polynomial sense.

This appendix's technical details mainly rely on standard conclusions from traditional computational complexity and quantum information, what this paper gives at geometric level is only structural translation and unification of these conclusions within computational universe framework.