Concept Sketching for Description Logics

Jinghan Wu[1,2] **Chang Lu**[3] **Renate A. Schmidt**[4] **Yizheng Zhao***[1,2]

[1]State Key Laboratory of Novel Software Technology, Nanjing University, Nanjing, China
[2]School of Artificial Intelligence, Nanjing University, Nanjing, China
[3]Computational Biology and Biomedical Informatics Program, Yale University, New Haven, CT, USA
[4]Department of Computer Science, The University of Manchester, Manchester, UK

Abstract

Domain experts often know *part* of a target description logic (DL) concept (a conjunction involving an existential restriction here, a structural pattern there), yet every existing concept learner ignores this knowledge and searches from scratch. We introduce *concept sketching* for the DL $\mathcal{ALC}$: the expert supplies a partial formula with typed holes (unknown subconcepts, unary or binary constructors), and any suitable back-end learner completes it from labeled examples. This *sketch fitting* problem turns out to have a sharp complexity landscape governed by a single dividing line: the signature size $|\Sigma|$. The problem is NP-complete even for a single hole; with $|\Sigma|$ as parameter it becomes fixed-parameter tractable; but without $|\Sigma|$ it is W[2]-hard, except for Type-2-only sketches which remain tractable. This dichotomy parallels recent results in first-order and decision tree learning. On the statistical side, we prove that each hole type contributes independently to PAC sample complexity, with a matching lower bound, so that a sketch capturing even part of the target structure requires provably fewer examples than unrestricted bounded fitting. Experiments on five standard DL benchmarks with two fundamentally different back-ends (SAT-based fitting and evolutionary concept learning) confirm improved accuracy, sample efficiency, and solver speed, demonstrating that sketching benefits are learner-agnostic.

1 INTRODUCTION

Learning description logic (DL) concepts [Lehmann and Hitzler, 2010] from labeled examples is a central task in ontology engineering and knowledge graph analysis. Given positive and negative examples drawn from a knowledge base, the goal is to construct a concept that correctly classifies the data and generalizes to unseen instances. This task has been approached from several angles. *Refinement-based* learners such as CELOE [Lehmann et al., 2011] and the broader DL-Learner framework [Bühmann et al., 2016] navigate the concept space top-down via refinement operators guided by heuristic scoring, building on early inductive logic programming for DLs [Fanizzi et al., 2008, Lehmann and Haase, 2009]. *Evolutionary* methods such as EvoLearner [Heindorf et al., 2022] evolve candidate concepts, represented as abstract syntax trees, through crossover and mutation, with the population initialized via biased random walks over the knowledge graph. *Neural* approaches include NCES [Kouagou et al., 2023, 2024], which frames concept learning as sequence-to-sequence synthesis, and DRILL [Demir and Ngomo, 2023], which uses deep reinforcement learning to guide refinement search. Finally, *SAT-based* methods encode bounded fitting as satisfiability problems: SPELL [ten Cate et al., 2023] learns $\mathcal{EL}$-concepts under $\mathcal{ELH}^r$-ontologies, and was recently extended to $\mathcal{ALC}$ [Funk et al., 2025] (hereafter SPELL-$\mathcal{ALC}$). Standard benchmarks from SML-Bench [Westphal et al., 2019] allow these methods to be compared systematically. For all their differences, they share one assumption, which we examine next.

That shared assumption is full automation. *Every* existing DL concept learner explores the concept space from scratch, with no mechanism for a domain expert to supply structural knowledge about the target concept. Yet in practice, experts frequently possess such knowledge. An ontology engineer may know that the target concept involves an existential restriction over a particular role but be uncertain about the filler; a knowledge engineer debugging a faulty class definition [Baader et al., 2007] may wish to preserve the overall structure while replacing specific subconcepts; a domain analyst may know the answer "involves a conjunction of two conditions" without knowing the conditions themselves. In each scenario, the expert has a *partial specification* (a

*Corresponding author.

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:7468–7483.

formula with blanks to be filled) that substantially constrains the concept space.

The idea of completing partial specifications is, however, well established in neighboring fields. Solar-Lezama's *program sketching* [Solar-Lezama et al., 2006, Solar-Lezama, 2013, Solar-Lezama, 2008] introduced the paradigm of partial programs with *holes* completed via counterexample-guided inductive synthesis, later formalized through the syntax-guided synthesis framework [Alur et al., 2013] and the oracle-guided perspective of Jha and Seshia [2017]. Most directly relevant to our setting, Lutz et al. [2023] transferred this paradigm to temporal logic: their *specification sketching for LTL* lets users provide partial linear temporal logic formulas with placeholder holes, and completes them from example traces via SAT solving. This line has since grown in two directions: the sketching and learning of formulas has been carried to branching-time logics such as CTL and ATL [Bordais et al., 2024, 2025], and constrained variants have been studied for LTL itself [Zhang et al., 2025], alongside a broader literature on learning temporal formulas from traces [Neider and Gavran, 2018, Gaglione et al., 2021, Neider and Roy, 2024]. Across this body of work, however, the complexity of completion has been settled only at the classical level, membership in NP; neither its parameterized complexity nor its sample complexity has been examined.

A related but distinct line is unification in DLs [Baader and Narendran, 2001, Baader and Morawska, 2010], where one seeks substitutions for concept variables that make concept expressions equivalent. This too involves filling in unknowns, but the objective is equational rather than supervised. Concept sketching accepts a completion when it separates labeled positive and negative examples, so the syntactic resemblance does not transfer the model-theoretic or learning-theoretic analysis.

The success of specification sketching in temporal logic raises a natural question: *can the same paradigm be transferred to DL concept learning?* We adopt the typed-hole viewpoint, but the DL setting changes what has to be analyzed. Both DL concepts and LTL formulas are tree-structured, and both admit SAT-based bounded fitting. Yet the transfer is nontrivial: role name selection in quantified subconcepts introduces a large discrete choice absent in LTL, interaction with background ontologies may change computational complexity, and semantics is evaluated over graph-structured interpretations rather than linear traces. Beyond these, while LTL sketching has been analyzed only in terms of classical (NP) complexity, the DL setting, with its richer signature structure, demands a *parameterized* complexity analysis.

The naïve approach to sketch completion, enumerating all possible completions, has a running time that depends critically on the number of available symbols. Each Type-0 hole (unknown subconcept of bounded size b) admits $|\Sigma|^{O(b)}$ candidates, where Σ is the signature. When $|\Sigma|$ is treated as part of the parameter, the total enumeration is a function of the parameter alone times a polynomial in the input size, the hallmark of *fixed-parameter tractability* (FPT) [Downey and Fellows, 2013, Cygan et al., 2015]. But when $|\Sigma|$ grows with the input, the polynomial's exponent depends on the parameter, giving only membership in XP. Whether this gap is inherent, that is, whether FPT is ruled out under standard assumptions, requires establishing *hardness* for the W-hierarchy.

Analogous dichotomies have been identified in related learning problems. van Bergerem et al. [2022] proved that learning first-order definable concepts is FPT over nowhere dense (sparse) structures but $\mathrm{AW}[*]$-hard in general, a dichotomy recently extended to monadic second-order logic [van Bergerem et al., 2025]. Ordyniak and Szeider [2021] established W-hardness of minimum decision tree learning via HITTING SET, a technique later refined by Gahlawat and Zehavi [2024]. Brand et al. [2023] and Ganian et al. [2023] further developed a general framework connecting parameterized consistency checking to PAC learnability. A common pattern runs through these results: in each case a single structural parameter separates the tractable from the intractable regime, whether it is signature size, structural sparsity, or the size of the feature space. Concept sketching, we will see, is no exception.

The complexity question tells us *when* sketching is computationally tractable; the complementary statistical question is *how many examples* it saves. The value of restricting the hypothesis class is well understood through probably approximately correct (PAC) learning [Valiant, 1984]: a consistent learner over a finite class H needs $O(\varepsilon^{-1}(\log|H| + \log\delta^{-1}))$ examples [Blumer et al., 1989]. The learnability of DL concepts has a long history, from PAC-learnable fragments of the early DL CLASSIC [Cohen and Hirsh, 1994] to exact learning with membership and equivalence queries [Frazier and Pitt, 1996, ten Cate and Dalmau, 2022, Angluin, 1988], including active and exact learning of $\mathcal{EL}$ and $\mathcal{ELI}$ concepts under ontologies [Funk et al., 2021, 2022]. The closest point of comparison for us is the SAT-based bounded fitting of ten Cate et al. [2023], whose sample complexity scales with $\|q_T\| \log|\Sigma|$, the target concept size times the log of the signature. A sketch that fixes part of the concept structure replaces the full concept space with a smaller *completion space* whose size depends on the holes, not the full target concept, yielding provably fewer examples.

Contributions. We introduce *concept sketching* for $\mathcal{ALC}$ and provide a theoretical and experimental analysis:

1. **Formalization.** We define $\mathcal{ALC}$ concept sketches with three typed placeholder holes, namely Type-0 (unknown subconcepts of bounded size), Type-1 (unknown unary constructors: $\neg$, $\exists r$, $\forall r$, or identity), and Type-2 (un-

known binary constructors: $\sqcap$ or $\sqcup$), and formalize the *sketch fitting problem*. Standard bounded fitting [ten Cate et al., 2023, Funk et al., 2025] is a special case ($\psi = ?_1^0$), so sketching smoothly interpolates between full automation and full specification.

2. **Parameterized complexity landscape.** Writing h for the total number of holes and b for the size bound on each Type-0 completion, we give a sharp picture governed by the signature: sketch fitting is NP-complete even for a single hole, becomes FPT when parameterized by $(h, b, |\Sigma|)$, but is W[2]-hard by (h, b) alone once $|\Sigma|$ is unbounded, with Type-2-only sketches remaining FPT by h_2. The signature dependence is thus inherent, not an artifact of enumeration.
3. **Sample complexity.** We prove that sketch fitting is a PAC learner [Valiant, 1984] with sample complexity

$$m = O\big(\varepsilon^{-1}\big(h_0 b \log|\Sigma| + h_1 \log(2|\Sigma_R|+2) + h_2 + \log\delta^{-1}\big)\big),$$

separately quantifying each hole type's contribution. This is substantially tighter than the unrestricted bound of ten Cate et al. [2023], which scales with $\|q_T\| \log|\Sigma|$ instead of $h_0 b \log|\Sigma|$, when the sketch fixes much of the target structure ($h_0 b \ll \|q_T\|$); Corollary 5.2 gives the precise condition, grounding the value of structural knowledge via Blumer et al. [1989]. We complement this with a matching lower bound, so the sample complexity of each hole type is optimal up to constant factors.
4. **Back-end integration.** We instantiate concept sketching with two back-ends. First, we extend the SPELL-$\mathcal{ALC}$ syntax tree encoding [Funk et al., 2025] to handle sketch constraints, reducing the number of free variables. Second, we integrate sketches into EvoLearner [Heindorf et al., 2022] by seeding the initial population with sketch completions and restricting mutation to hole positions. On standard benchmarks, both back-ends benefit from sketches: accuracy, sample efficiency, and search speed all improve.

2 PRELIMINARIES

A *signature* is a pair $\Sigma = (\Sigma_C, \Sigma_R)$ of finite, disjoint sets of *concept names* $A \in \Sigma_C$ and *role names* $r \in \Sigma_R$; we write $|\Sigma| = |\Sigma_C| + |\Sigma_R|$. Throughout, we assume $\Sigma_C \neq \emptyset$; role names may be absent ($\Sigma_R = \emptyset$). $\mathcal{ALC}$ *concepts* over Σ are formed by:

$$C, D \;::=\; \top \mid \bot \mid A \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \exists r.\, C \mid \forall r.\, C\,,$$

where $A \in \Sigma_C$ and $r \in \Sigma_R$. The operators split into *unary constructors* $\neg$, $\exists r$, $\forall r$ and *binary constructors* $\sqcap$, $\sqcup$. The *size* $\|C\|$ of a concept C is the number of nodes in its syntax tree (e.g., $\|A \sqcap \exists r.\, \neg B\| = 5$). For a size bound k, we write $\mathcal{L}_{\Sigma,k}$ for the set of all $\mathcal{ALC}$ concepts over Σ of size at most k.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty finite *domain* $\Delta^{\mathcal{I}}$ and a function mapping each $A \in \Sigma_C$ to $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and each $r \in \Sigma_R$ to $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The *extension* $C^{\mathcal{I}}$ is defined inductively in the standard way [Baader et al., 2017]: in particular, $(\exists r.\, C)^{\mathcal{I}}$ collects all a with an r-successor in $C^{\mathcal{I}}$, and $(\forall r.\, C)^{\mathcal{I}}$ collects all a whose every r-successor lies in $C^{\mathcal{I}}$. We write $a \in C^{\mathcal{I}}$ and say a is an *instance* of C in $\mathcal{I}$.

A *labeled example* is a pair $(\mathcal{I}, a)$ with $a \in \Delta^{\mathcal{I}}$. A concept C *fits* examples (P, N) if C accepts (i.e., $a \in C^{\mathcal{I}}$) every $(\mathcal{I}, a) \in P$ and rejects every $(\mathcal{I}, a) \in N$. We assume that P and N are disjoint, since each example carries exactly one label. Empty sides are allowed: if $P = \emptyset$, then $\bot$ fits all examples, and if $N = \emptyset$, then $\top$ fits all examples.

Definition 2.1 (Bounded fitting)**.** Given labeled examples (P, N), a signature Σ, and a size bound $s \in \mathbb{N}$, the *bounded fitting* problem asks: does there exist an $\mathcal{ALC}$ concept C over Σ with $\|C\| \leq s$ that fits (P, N)?

Bounded fitting for $\mathcal{ALC}$ is NP-complete [Funk et al., 2025]. NP membership relies on the fact that, for finite interpretations without a TBox, instance checking is polynomial in $|\mathcal{I}|$ and $\|C\|$.

Remark 2.2. We follow the ontology-free setting of SAT-based $\mathcal{ALC}$ bounded fitting [Funk et al., 2025]: examples are pairs $(\mathcal{I}, a)$ with $\mathcal{I}$ given explicitly, and negation is interpreted as complement over its finite domain. Thus our setting is closed-world over the given finite interpretation, rather than open-world reasoning; moreover, we do not include TBox-mediated reasoning. With a general $\mathcal{ALC}$ TBox, instance checking is ExpTime-complete [Baader et al., 2017]; extending sketching to such settings is left as future work.

For PAC learning [Valiant, 1984], a learner receives m i.i.d. examples from an unknown distribution, labeled by a target q_T, and must output a hypothesis with error $\leq \varepsilon$ with probability $\geq 1-\delta$. The classical finite-class bound [Blumer et al., 1989] gives:

Proposition 2.3 (Occam bound)**.** *Let $\mathcal{C}$ be a finite concept class with $|\mathcal{C}| \geq 2$. Any consistent learner (ε, δ)-learns $\mathcal{C}$ from $m \geq \varepsilon^{-1}(\log|\mathcal{C}| + \log\delta^{-1})$ examples.*

3 CONCEPT SKETCHES

We extend the $\mathcal{ALC}$ syntax with three types of placeholder holes, each occupying a distinct syntactic position, and formalize the problem of completing them from labeled data.

Definition 3.1 (Concept sketch)**.** Let $\Sigma = (\Sigma_C, \Sigma_R)$ be a signature and let $b \in \mathbb{N}$ be a *hole size bound*. A *concept*

sketch over (Σ, b) is a formula generated by the grammar

$$\psi, \varphi \;::=\; \top \mid \bot \mid A \mid \neg\psi \mid \psi \sqcap \varphi \mid \psi \sqcup \varphi \mid \exists r.\, \psi \mid \forall r.\, \psi \\ \mid ?_i^0 \mid ?_j^1(\psi) \mid \psi\ ?_k^2\ \varphi\,,$$

where $A \in \Sigma_C$, $r \in \Sigma_R$, and i, j, k are distinct indices, subject to the constraint that every hole index appears exactly once in ψ. The three hole types are:

- **Type-0** (unknown subconcept): $?_i^0$ is replaced by a concept $C \in \mathcal{L}_{\Sigma,b}$ of size at most b.
- **Type-1** (unknown unary constructor): $?_j^1(\psi)$ is replaced by one of $U_\Sigma = \{\mathrm{id},\ \neg,\ \exists r,\ \forall r \mid r \in \Sigma_R\}$, where id denotes the identity (the hole is removed and its argument kept).
- **Type-2** (unknown binary constructor): $\psi\ ?_k^2\ \varphi$ is replaced by $\sqcap$ or $\sqcup$.

We write h_0, h_1, h_2 for the number of holes of each type and $h = h_0 + h_1 + h_2$ for the total.

The *fixed part* of a sketch consists of all non-hole nodes. A sketch with $h = 0$ is an ordinary $\mathcal{ALC}$ concept. The *size* $\|\psi\|$ is defined as for concepts, with each hole marker counting as a single node. Including the identity in the Type-1 domain lets the expert express uncertainty about *whether* a unary constructor is present, not only *which* one: e.g., $?_1^1(\exists r.\, A)$ can resolve to $\neg\exists r.\, A$ or simply $\exists r.\, A$.

A *completion* σ maps each Type-0 hole $?_i^0$ to a concept $\sigma(?_i^0) \in \mathcal{L}_{\Sigma,b}$, each Type-1 hole $?_j^1$ to an element $\sigma(?_j^1) \in U_\Sigma$ ($|U_\Sigma| = 2|\Sigma_R|{+}2$), and each Type-2 hole $?_k^2$ to a constructor $\sigma(?_k^2) \in \{\sqcap, \sqcup\}$. The *completed concept* $\psi[\sigma]$ is obtained by simultaneously replacing every hole with its assigned value. The *completion space* $\mathcal{C}_\psi = \{\, \psi[\sigma] \mid \sigma \text{ is a completion of } \psi \,\}$ collects all concepts obtainable from ψ.

Lemma 3.2 (Completion space size)**.** $|\mathcal{C}_\psi| \le |\mathcal{L}_{\Sigma,b}|^{h_0} \cdot (2|\Sigma_R| + 2)^{h_1} \cdot 2^{h_2}$.

Proof. Holes are filled independently; the bound follows by multiplication of domain sizes. It is an upper bound because distinct completions may yield the same concept. □

Lemma 3.3 (Concept language size)**.** $|\mathcal{L}_{\Sigma,b}| \le (4\kappa)^{b+1}$ *where* $\kappa = |\Sigma_C| + 2|\Sigma_R| + 5$. *In particular,* $\log|\mathcal{L}_{\Sigma,b}| = O(b \log|\Sigma|)$ *for* $|\Sigma| \ge 2$.

Proof. An $\mathcal{ALC}$ concept of size i corresponds to a rooted ordered tree on i nodes in which every node has $\le \kappa$ label options. The number of such trees is $\le C_i \cdot \kappa^i \le (4\kappa)^i$, where $C_i \le 4^i$ is the i-th Catalan number. Summing over $i = 1, \ldots, b$ gives $\sum_{i=1}^{b} (4\kappa)^i \le (4\kappa)^{b+1}$ since $4\kappa \ge 20$. The logarithmic bound follows from $\kappa = |\Sigma_C| + 2|\Sigma_R| + 5 \le 2|\Sigma| + 5$. □

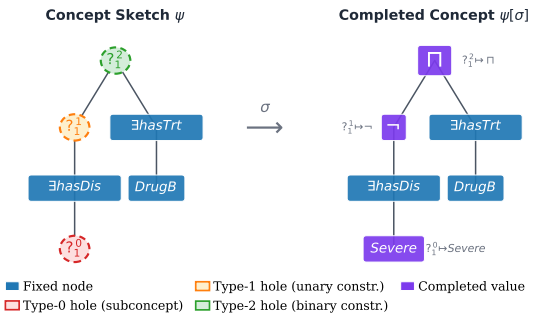

Figure 1: A sketch (left) with one hole of each type: Type-0 (red), Type-1 (orange), Type-2 (green); and the completed concept (right). Fixed nodes are blue.

Combining Lemmas 3.2 and 3.3 yields:

$$\log|\mathcal{C}_\psi| = O(h_0 b \log|\Sigma| + h_1 \log(2|\Sigma_R|{+}2) + h_2).$$

Definition 3.4 (Sketch fitting)**.** Given a concept sketch ψ over (Σ, b) and labeled examples (P, N), the *sketch fitting* problem asks: does there exist a completion σ of ψ such that $\psi[\sigma]$ fits (P, N)?

Two boundary cases connect sketch fitting to existing problems. When $\psi = ?_1^0$ (a single Type-0 hole with $b = s$), sketch fitting reduces to bounded fitting (Definition 2.1). When $h = 0$, it reduces to polynomial-time model checking. Thus sketch fitting *interpolates* between fully automated concept learning and fully specified verification; every fixed node narrows the search space.

Figure 1 illustrates a sketch with one hole of each type and its completion. For a simpler numeric illustration, take the target $q_T = \exists \mathit{hasDis}.\, \mathit{Severe} \sqcap \exists \mathit{hasTrt}.\, \mathit{DrugB}$. An expert knowing the $\sqcap$ structure but not the fillers writes $\psi = \exists \mathit{hasDis}.\, ?_1^0 \sqcap \exists \mathit{hasTrt}.\, ?_2^0$ ($h_0{=}2$, $b{=}1$). The completion space has $|\mathcal{L}_{\Sigma,1}|^2 = 64$ elements, whereas unrestricted bounded fitting with $s = 5$ searches $|\mathcal{L}_{\Sigma,5}| = 31{,}592$ candidates, roughly 500 times larger.

4 PARAMETERIZED COMPLEXITY

A *parameterized problem* is a language $L \subseteq \{0,1\}^* \times \mathbb{N}$ with parameter k. It is *fixed-parameter tractable* (FPT) if solvable in $f(k) \cdot |x|^{O(1)}$ time, and belongs to XP if solvable in $|x|^{g(k)}$ time. The *W-hierarchy* FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ XP stratifies the gap: W[t]-hardness gives strong evidence against FPT membership [Downey and Fellows, 2013, Cygan et al., 2015].

Theorem 4.1 (NP-completeness)**.** *Sketch fitting is NP-complete, even only for a single hole.*

Proof. Membership. Guess a completion σ; its description is polynomial in $\|\psi\|$ and b (each Type-0 value is a concept

of size $\leq b$, each Type-1 value is one of $2|\Sigma_R|+2$ constructors, each Type-2 value is $\sqcap$ or $\sqcup$). Verifying $\psi[\sigma]$ fits (P, N) takes polynomial time via instance checking. *Hardness.* Bounded fitting is NP-hard [Funk et al., 2025] and is the special case $\psi = ?_1^0$. □

Theorem 4.2 (Fixed-parameter tractability)**.** *Sketch fitting parameterized by $(h, b, |\Sigma|)$ is FPT.*

Proof. Enumerate all completions. By Lemmas 3.2 and 3.3, the count is bounded by $(4\kappa)^{(b+1)h_0} \cdot (2|\Sigma_R|+2)^{h_1} \cdot 2^{h_2}$, which depends only on the parameter $(h, b, |\Sigma|)$. Verifying each takes polynomial time, giving total time $f(h, b, |\Sigma|) \cdot \mathrm{poly}(|P|+|N|+\|\psi\|)$. □

Remark 4.3. Without $|\Sigma|$ in the parameter, sketch fitting is still in XP: since $|\mathcal{L}_{\Sigma,b}| \leq (4\kappa)^{b+1}$ and $\kappa = O(|\Sigma|)$, the Type-0 enumeration contributes $|\Sigma|^{O(h_0 b)}$; Type-1 and Type-2 contribute $|\Sigma|^{O(h_1)}$ and 2^{h_2}, giving total time $|\Sigma|^{O(hb)} \cdot \mathrm{poly}(n)$. Theorem 4.4 shows that this cannot be improved to FPT unless W[2] = FPT. Pinning down the exact class between W[2] and XP is open: the W[2]-hardness is driven by the choice of concept names, whereas any completion must still be verified by propagating semantics through the unboundedly deep quantifier nesting of the sketch, so membership in W[2] is not evident.

Theorem 4.4 (W[2]-hardness)**.** *Sketch fitting parameterized by (h, b) is W[2]-hard, even when restricted to Type-0-only sketches with $b = 1$ and $\Sigma_R = \emptyset$.*

Proof sketch (full proof in Appendix A). We reduce from HITTING SET [Downey and Fellows, 2013], which is W[2]-complete parameterized by solution size k. Given a universe $U = \{u_1, \ldots, u_n\}$ and subsets $\mathcal{F} = \{S_1, \ldots, S_m\}$, set $\Sigma_C = U$, $\Sigma_R = \emptyset$, and define the sketch $\psi = ?_1^0 \sqcup \cdots \sqcup ?_k^0$ (a k-ary disjunction of Type-0 holes with $b = 1$). For each $S_j \in \mathcal{F}$, construct an interpretation $\mathcal{I}_j$ with domain $\{a_j, d_j\}$ where $u_i^{\mathcal{I}_j} = \{a_j\}$ if $u_i \in S_j$ and $\emptyset$ otherwise; the positive example is $(\mathcal{I}_j, a_j)$ and the negative is $(\mathcal{I}_j, d_j)$.

A *monotonization lemma* shows that any fitting completion can be assumed to use only concept names (neither $\top$ nor $\bot$): $\top$ violates all negative examples, $\bot$ contributes nothing and can be replaced. With this, the completions correspond exactly to subsets of U of size k that hit every S_j, i.e., hitting sets. The parameter maps as $(h, b) = (k, 1)$. □

The hardness employs only Σ_C (no role names), confirming that the concept name space drives intractability, matching the dichotomies in first-order learning [van Bergerem et al., 2022, 2025], decision tree learning [Ordyniak and Szeider, 2021, Gahlawat and Zehavi, 2024], and parameterized consistency checking [Brand et al., 2023, Ganian et al., 2023]. The reduction works because the role-free setting collapses $\mathcal{ALC}$ to propositional logic, where concept names act as Boolean atoms, so that choosing k concept names to satisfy every example is exactly HITTING SET.

Table 1: Parameterized complexity of sketch fitting.

Parameterization	Result	Reference
(none)	NP-complete	Theorem 4.1
$(h, b, \|\Sigma\|)$	FPT	Theorem 4.2
(h, b)	XP	Remark 4.3
(h, b)	W[2]-hard	Theorem 4.4
h_2 (Type-2 only)	FPT	Theorem 4.5

Theorem 4.5 (Type-2 sketches)**.** *For sketches with $h_0 = h_1 = 0$, sketch fitting is FPT parameterized by h_2 alone.*

Proof. Each Type-2 hole has domain $\{\sqcap, \sqcup\}$, so the total number of completions is 2^{h_2}, independent of $|\Sigma|$. Enumerate and verify each in polynomial time. □

The key observation is that Type-2 holes have a constant-size domain; the W[2]-hardness originates from Type-0 holes whose domain scales with $|\Sigma|$. We leave the complexity of Type-1-only sketches as an open question (Section 8). Table 1 summarizes the landscape.

5 SAMPLE COMPLEXITY

Fix a sketch ψ over (Σ, b) with completion space $\mathcal{C}_\psi$ (Definition 3.1). A sketch fitting solver is a consistent learner over $\mathcal{C}_\psi$: it returns a completion whose induced concept fits every example. Under the *realizability assumption* $q_T \in \mathcal{C}_\psi$, the Occam bound (Proposition 2.3) yields:

Theorem 5.1 (Sample complexity)**.** *Let ψ be a concept sketch with h_0 Type-0, h_1 Type-1, and h_2 Type-2 holes. Assume $q_T \in \mathcal{C}_\psi$. Then sketch fitting (ε, δ)-learns $\mathcal{C}_\psi$ from*

$$m \;=\; O\Big(\varepsilon^{-1}\big(h_0\, b \log|\Sigma| + h_1 \log(2|\Sigma_R|+2) \;+\; h_2 + \log\delta^{-1}\big)\Big) \tag{1}$$

labeled examples.

Proof. If $h = 0$ then $\mathcal{C}_\psi = \{\psi\}$ and realizability forces $\psi = q_T$, so zero examples suffice and (1) holds trivially; assume $h \geq 1$. The Occam bound (Proposition 2.3) requires $|\mathcal{C}_\psi| \geq 2$; this holds because every hole domain has size at least 2 ($\{\sqcap, \sqcup\}$ for Type-2, $\{\top, \bot\}$ for Type-0, etc.). By Lemma 3.2,

$$\log|\mathcal{C}_\psi| \;\leq\; h_0 \log|\mathcal{L}_{\Sigma,b}| + h_1 \log(2|\Sigma_R|+2) + h_2 \log 2.$$

Lemma 3.3 gives $\log|\mathcal{L}_{\Sigma,b}| = O(b \log|\Sigma|)$. Substituting into Proposition 2.3 yields (1). □

Each hole type contributes an independent term, so the expert's structural knowledge is credited separately: Type-0 holes are the most expensive ($O(b \log |\Sigma|)$ each, the full concept space per subconcept), Type-1 holes cost only $O(\log(2|\Sigma_R|+2))$ (they range over role names, not concept names), and Type-2 holes are a constant each. Every fixed node the expert provides shrinks the hypothesis space.

Unrestricted bounded fitting is the special case $\psi = ?_1^0$ with $b = \|q_T\|$, giving the bound $m_{\text{full}} = O(\varepsilon^{-1}(\|q_T\| \log |\Sigma| + \log \delta^{-1}))$, recovering ten Cate et al. [2023].

Corollary 5.2. *Let $|\Sigma| \geq 2$. Whenever $h_0\, b + 2h_1 + h_2 < \|q_T\|$, i.e., the sketch fixes at least one node of the target (with each Type-1 hole charged as two nodes), the sketch bound* (1) *is strictly tighter than bounded fitting.*

Proof sketch (full proof in Appendix D). Bounding $h_1 \log(2|\Sigma_R|+2) \leq 2h_1 \log |\Sigma|$ (using $\Sigma_C \neq \emptyset$ and $|\Sigma| \geq 2$) and $h_2 \leq h_2 \log |\Sigma|$, the sketch bound is at most $(h_0 b + 2h_1 + h_2) \log |\Sigma| + \log \delta^{-1}$, which the condition makes strictly smaller than the bounded fitting bound. The coefficient 2 on h_1 is tight. □

The realizability assumption $q_T \in \mathcal{C}_\psi$ requires the sketch to be "correct"; relaxing this assumption is an interesting future direction. The earlier example ($h_0 = 2$, $b = 1$, $|\Sigma| = 8$, $\|q_T\| = 5$) illustrates the savings: the sketch bound is proportional to $h_0 b \log |\Sigma| = 6$ versus $\|q_T\| \log |\Sigma| = 15$, a $2.5\times$ reduction that grows with richer signatures.

The upper bound of Theorem 5.1 is optimal up to constant factors, in the following worst-case sense.

Theorem 5.3 (Matching lower bound)**.** *There is a family of concept sketches, with h_0 Type-0, h_1 Type-1, and h_2 Type-2 holes over a signature Σ, for which any algorithm that (ε, δ)-learns $\mathcal{C}_\psi$ requires*

$$m = \Omega\Big(\varepsilon^{-1}\Big(h_0\, b \log |\Sigma| + h_1 \log(2|\Sigma_R|+2) + h_2 + \log \delta^{-1}\Big)\Big)$$

examples.

The proof exhibits, for each hole type, an explicit set of examples shattered by the completions of a single hole: a Type-0 hole ranging over size-b conjunctions of concept names shatters $\Theta(b \log |\Sigma|)$ points (the classical Vapnik-Chervonenkis dimension of bounded-size monotone conjunctions), a Type-1 hole shatters $\Theta(\log(2|\Sigma_R|+2))$ points by encoding role indices in binary, and each Type-2 hole toggles one dedicated example. Placing the three gadgets in parallel and invoking the general sample-complexity lower bound for a class of Vapnik-Chervonenkis dimension d gives the claim; the full construction is in Appendix E. Since the exhibited family attains the upper bound of Theorem 5.1, that bound cannot be improved in general, so each hole type is priced at its worst-case-optimal rate.

6 SAT-BASED SKETCH FITTING

The FPT algorithm of Theorem 4.2 solves sketch fitting by brute-force enumeration; for practical performance, we encode the problem as a propositional satisfiability instance following the SAT-based SPELL-$\mathcal{ALC}$ [Funk et al., 2025]. As in Remark 2.2, the encoding assumes examples of the form $(\mathcal{I}, a)$ with $\mathcal{I}$ given as an explicit finite interpretation, and does not perform TBox-mediated reasoning.

The baseline encoding represents a candidate concept of size $\leq s$ as a syntax tree T with s nodes. Each node v has a *constructor variable* $\mu_{v,c}$ for every possible operator c ($|\Sigma_C|+2|\Sigma_R|+5$ per node; one-hot constrained) and a *semantics variable* $\eta_{v,j,d}$ for each interpretation j and domain element d, encoding $d \in C_v^{\mathcal{I}_j}$. Propagation clauses enforce $\mathcal{ALC}$ semantics; fitting clauses enforce acceptance of positive and rejection of negative examples. See Funk et al. [2025] for the base encoding; our sketch encoding fixes the tree topology given by ψ, so the formula is polynomial in s, $|\Sigma|$, and $|\Delta|$, where $|\Delta| = \sum_j |\Delta^{\mathcal{I}_j}|$.

To exploit sketch structure, we modify the encoding so that each node of ψ falls into one of four categories: (i) *Fixed nodes*: the known constructor is set via a unit clause, which SAT preprocessing eliminates. (ii) *Type-0 holes*: each hole is expanded into a free subtree of size b, encoded exactly as in the baseline. (iii) *Type-1 holes*: the constructor variables are restricted to unary constructors plus identity ($2|\Sigma_R|+2$ choices); the identity propagation rule $\eta_{v,j,d} \Leftrightarrow \eta_{v_c,j,d}$ handles the id case. (iv) *Type-2 holes*: only the two constructor variables $\mu_{v,\sqcap}$ and $\mu_{v,\sqcup}$ remain, reducing the node to a single Boolean.

Theorem 6.1 (Encoding correctness)**.** *The sketch-constrained formula Φ_ψ is satisfiable if and only if there exists a fitting completion.*

Proof sketch (full proof in Appendix B). ($\Rightarrow$) Given a satisfying assignment τ, read off σ: for each Type-0 hole, the one-hot and propagation clauses on the free subtree determine a unique concept in $\mathcal{L}_{\Sigma,b}$; for each Type-1 hole, the unique active variable among the $2|\Sigma_R|+2$ choices determines the constructor (with id handled by the identity propagation rule above); for each Type-2 hole, the single Boolean determines $\sqcap$ vs. $\sqcup$. The semantics variables then propagate correctly through the entire tree, so the fitting clauses ensure $\psi[\sigma]$ fits (P, N). ($\Leftarrow$) Given σ, set constructor variables per hole type, and define $\eta_{v,j,d}$ by evaluating $\psi[\sigma]$; all clauses are satisfied because they encode $\mathcal{ALC}$ semantics. □

Proposition 6.2 (Variable reduction)**.** *Recall $\kappa = |\Sigma_C| + 2|\Sigma_R| + 5$ and let f be the number of fixed nodes. The sketch encoding uses $h_0 b\kappa + h_1(2|\Sigma_R|+2) + h_2$ free constructor variables, compared to $(f + h_0 b + h_1 + h_2)\kappa$ in the unrestricted encoding on the same tree, a reduction of $f\kappa + h_1(\kappa - 2|\Sigma_R| - 2) + h_2(\kappa - 1)$.*

Proof sketch (full proof in Appendix C). Counting free constructor variables per node type: fixed nodes contribute none (eliminated by unit propagation), Type-0 subtree nodes contribute κ each, and Type-1 and Type-2 holes contribute $2|\Sigma_R|+2$ and 1 respectively. Summing the per-node savings gives the stated reduction. □

The free variable count has the same structure as $\log|\mathcal{C}_\psi|$, reinforcing that the sketch eliminates the same choices from both the statistical and computational perspectives.

7 EXPERIMENTS

We evaluate concept sketching on five standard DL learning benchmarks, addressing four questions: (Q1) Does sketch structure improve accuracy? (Q2) Do sketches reduce sample requirements, as predicted by Theorem 5.1? (Q3) Does the sketch-constrained encoding speed up solving? (Q4) Are the benefits learner-agnostic, i.e., do they transfer from SAT-based fitting to evolutionary concept learning?

7.1 SETUP

Benchmarks and Baselines. We evaluate on five standard DL learning benchmarks (Carcinogenesis, Mutagenesis, Hepatitis, Mammographics, and Family) whose statistics are summarized in Table 6 (Appendix G). Following Remark 2.2, we use only ABox assertions. We measure classification performance by *example-level* F1-score. We compare five methods: **BoundedFit** (unrestricted, $\alpha = 0$, equivalent to SPELL-$\mathcal{ALC}$ [Funk et al., 2025]), **SketchFit** (our sketch-constrained solver) at varying α, **EvoLearner** [Heindorf et al., 2022] (evolutionary baseline without sketch), **Evo + Sketch** (EvoLearner with sketch-constrained initialization and mutation), and **CELOE** [Bühmann et al., 2016] (refinement-based baseline). BoundedFit (i.e., SPELL-$\mathcal{ALC}$ without sketch constraints) is the state-of-the-art SAT-based $\mathcal{ALC}$ concept learner and serves as our primary baseline; SketchFit extends its encoding with sketch constraints and is our proposed method. EvoLearner demonstrates that sketch benefits transfer to a non-SAT paradigm; CELOE serves as an external reference. Appendix F provides a systematic comparison of all existing DL concept learners and justifies these choices. For both BoundedFit and SketchFit, the size bound is set to $s = \|q_T\|$ (the target concept size listed in Table 6); this is oracle knowledge, consistent with the standard evaluation protocol of Funk et al. [2025].

Sketch generation. Since no benchmark supplies expert-authored sketches, we simulate expert knowledge by generating sketches from the known target q_T: define the *retention fraction* $\alpha \in [0, 1]$ as the fraction of q_T's syntax-tree nodes that are fixed. We evaluate $\alpha \in \{0, 0.25, 0.50, 0.75\}$; for $\alpha > 0$, unfixed nodes are replaced by holes typed according to their arity in q_T (leaves → Type-0 with $b = 1$; unary nodes → Type-1; binary nodes → Type-2), so the size bound within each Type-0 hole is always $b = 1$, while $\alpha = 0$ is the unrestricted BoundedFit baseline (a single Type-0 hole with $b = \|q_T\|$): a target-driven protocol that derives sketches from known targets, analogous to the evaluation of Lutz et al. [2023]. For each (dataset, α) pair, we generate 10 random sketches and report mean and standard deviation over 10 folds × 10 sketches = 100 evaluations. All experiments use stratified 10-fold cross validation, CaDiCaL [Biere et al., 2024] as the SAT solver for BoundedFit and SketchFit, and ontolearn 0.7[1] for EvoLearner, with a 300 s timeout per fold on a single thread of an AMD EPYC 7763 CPU.

This protocol isolates the effect of structural information, since current benchmarks do not provide expert-authored sketches. In practice, sketches may come from partial class definitions, domain templates, or expert-specified constraints on likely roles and constructors. Evaluating human-authored sketches is left to future work.

EvoLearner integration. For Evo + Sketch, we exploit the syntax tree representation of Heindorf et al. [2022]: the initial population is seeded with $N_{\text{pop}} = 100$ random completions of the sketch (each hole filled uniformly from its domain), replacing the default random-walk initialization. Crossover and mutation are restricted to *hole positions*: crossover may swap subtrees only at Type-0 holes, and point mutations may alter the constructor only at Type-1 or Type-2 holes; fixed nodes are protected. The fitness function, selection strategy, and termination criterion remain unchanged (tournament selection, $G_{\max} = 200$ generations, 300 s timeout). Vanilla EvoLearner uses the same hyperparameters and timeout for fair comparison; see Appendix G for details.

7.2 MAIN RESULTS (Q1, Q3)

Accuracy (Q1). Table 2 shows that SketchFit consistently outperforms BoundedFit, with the gap widening as α increases. On Carcinogenesis ($|\Sigma_C| = 142$), SketchFit at $\alpha = 0.75$ achieves F1 $= 0.78$, compared to 0.61 for BoundedFit and 0.71 for CELOE. The one exception is Mutagenesis, where CELOE has the highest F1 (we discuss why in Section 7.5); SketchFit has higher mean F1 than CELOE on the remaining four datasets. Evo + Sketch likewise outperforms vanilla EvoLearner on every dataset, with relative F1 gains of 5–13% at $\alpha = 0.75$. EvoLearner itself outperforms BoundedFit on four of five datasets, reflecting the complementary strengths of evolutionary search. The accuracy gains from sketching are consistent across both back-ends, supporting Q4.

Statistical significance. We tested the primary comparisons between each sketch-enhanced method and its non-

[1] `https://pypi.org/project/ontolearn/`

Table 2: F1-score (mean ± std). Bold: best per row.

Dataset	BndFit ($\alpha=0$)	Sketch ($\alpha=.25$)	Sketch ($\alpha=.50$)	Sketch ($\alpha=.75$)	Evo —	Evo+Sk ($\alpha=.75$)	CELOE
Carcinog.	$.61_{\pm.05}$	$.65_{\pm.06}$	$.71_{\pm.05}$	$\mathbf{.78}_{\pm.04}$	$.66_{\pm.07}$	$.74_{\pm.05}$	$.71_{\pm.01}$
Mutagen.	$.72_{\pm.08}$	$.78_{\pm.07}$	$.83_{\pm.06}$	$.89_{\pm.05}$	$.88_{\pm.06}$	$.92_{\pm.04}$	$\mathbf{.93}_{\pm.14}$
Hepatitis	$.58_{\pm.06}$	$.62_{\pm.07}$	$.67_{\pm.05}$	$\mathbf{.73}_{\pm.04}$	$.61_{\pm.06}$	$.69_{\pm.05}$	$.60_{\pm.02}$
Mammogr.	$.71_{\pm.05}$	$.74_{\pm.05}$	$.78_{\pm.04}$	$\mathbf{.83}_{\pm.04}$	$.69_{\pm.06}$	$.78_{\pm.04}$	$.64_{\pm.01}$
Family	$.82_{\pm.06}$	$.86_{\pm.05}$	$.90_{\pm.04}$	$\mathbf{.94}_{\pm.03}$	$.85_{\pm.05}$	$.91_{\pm.04}$	$.88_{\pm.05}$

Table 3: Free constructor variables (#Vars) and mean wall-clock time per fold (s); for EvoLearner, mean generations to convergence (Gen) and time.

	BndFit ($\alpha=0$)		Sketch ($\alpha=.50$)		Sketch ($\alpha=.75$)		Evo		Evo+Sk ($\alpha=.75$)	
Dataset	#Vars	Time	#Vars	Time	#Vars	Time	Gen	Time	Gen	Time
Carcinog.	1 510	187.4	680	52.1	295	21.3	168	142.7	72	58.1
Mutagen.	990	78.6	430	22.7	175	9.4	112	48.3	51	21.6
Hepatitis	290	12.3	140	4.8	65	2.1	95	35.1	42	14.8
Mammogr.	310	28.1	150	7.6	70	3.2	134	62.5	55	25.4
Family	410	31.5	185	9.3	80	3.8	108	41.2	38	16.5

sketch counterpart using paired Wilcoxon signed-rank tests on fold-level F1 scores (10 folds; for sketch methods, sketch instances were averaged within each fold). SketchFit at $\alpha = 0.75$ significantly outperforms BoundedFit on all five datasets, with ΔF1 ranging from $+0.12$ to $+0.17$ (all $p < 0.005$). Evo + Sketch at $\alpha = 0.75$ also outperforms vanilla EvoLearner on all five datasets, with ΔF1 ranging from $+0.04$ to $+0.09$; Mutagenesis is the weakest case (ΔF1 $= +0.04$, $p < 0.05$), while the other four datasets have $p < 0.005$. F1 increases monotonically with α on all five datasets for SketchFit, and Evo + Sketch improves over vanilla EvoLearner on all five.

Speedup (Q3). Table 3 shows that the sketch encoding dramatically reduces free constructor variables (77–82% at $\alpha = 0.75$), translating into roughly 6–9× speedup. On Carcinogenesis, BoundedFit averages 187.4 s per fold; SketchFit at $\alpha = 0.75$ solves in 21.3 s. For the evolutionary one (rightmost), Evo + Sketch converges in fewer generations than vanilla EvoLearner (38–72 vs. 95–168), yielding 2.2–2.5× wall-clock speedup. The mechanism differs, sketch-seeded initialization starts closer to the optimum rather than reducing problem size, but the practical effect is analogous.

7.3 SAMPLE EFFICIENCY (Q2)

We vary training set size $m \in \{5, 10, 20, 50, 100\}$ ($m \leq 50$ for Mutagenesis, which has only 84 examples): within each cross-validation fold, we draw m examples uniformly at random (without replacement) from the training split and evaluate on the full test split, averaging over 10 sketch instantiations × 5 random draws per m. Figure 2 shows that SketchFit achieves higher F1 than BoundedFit at every sample size, with the largest gains in the low-data regime: on Carcinogenesis with $m = 10$, SketchFit at $\alpha = 0.50$ reaches F1 $= 0.58$ versus 0.41 for BoundedFit, a 41% relative improvement, confirming Corollary 5.2: a smaller hypothesis class extracts more from each example. Evo + Sketch shows the same pattern: at $m = 10$ on Carcinogenesis, it reaches F1 $= 0.52$ versus 0.39 for vanilla EvoLearner, a 33% relative gain. The oracle ceiling ($\alpha = 1$) reveals substantial headroom on Carcinogenesis and Hepatitis, while Family and Mutagenesis curves nearly saturate at $\alpha = 0.75$, suggesting diminishing returns from additional structural knowledge on simpler targets.

Table 4: Hole type ablation on Carcinogenesis ($\alpha = 0.50$).

Hole type	F1	#Vars	Time (s)
Type-0 only	$.67_{\pm.06}$	820	63.4
Type-1 only	$.71_{\pm.05}$	140	11.2
Type-2 only	$.74_{\pm.03}$	5	0.4
Mixed (default)	$.71_{\pm.05}$	680	52.1

7.4 HOLE-TYPE ABLATION

We construct single-type sketches on Carcinogenesis at $\alpha = 0.50$ (Table 4). For each single-type variant, only nodes matching the designated arity are unfixed: e.g., "Type-0 only" unfixes only leaf nodes (replacing them by Type-0 holes with $b = 1$) while keeping all internal nodes fixed. Type-2-only sketches are solved fastest and with highest F1, while Type-0-only sketches are slowest, matching the theoretical cost hierarchy (Type-0 > Type-1 > Type-2) and the parameterized complexity results (Theorems 4.4 and 4.5).

7.5 DISCUSSION

The experiments confirm all theoretical predictions. Accuracy gains are most pronounced on large-signature datasets (Carcinogenesis: $|\Sigma_C| = 142$), where hypothesis space reduction is greatest, consistent with the $h_0 b \log |\Sigma|$ scaling in Theorem 5.1, and solver speedup tracks the variable reduction of Proposition 6.2. The hole type ablation provides particularly direct experimental validation of the parameterized complexity results: Type-2-only sketches are solved fastest (0.4 s), matching their FPT status (Theorem 4.5), while Type-0-only sketches are slowest (63.4 s), consistent with the W[2]-hardness originating from Type-0 holes (Theorem 4.4). On Family, even modest structural knowledge ($\alpha = 0.25$) suffices to match CELOE, because the target concept is relatively simple (small $\|q_T\|$) and the large role name set ($|\Sigma_R| = 15$) makes even partial fixation effective. The one exception to sketch dominance is Mutagenesis, where CELOE outperforms all sketch levels. This dataset has a comparatively small signature but shallow, regular target concepts that align with CELOE's top-down refinement; the small training set ($|E| = 84$) limits the statistical benefit of hypothesis space reduction. This suggests that sketch-

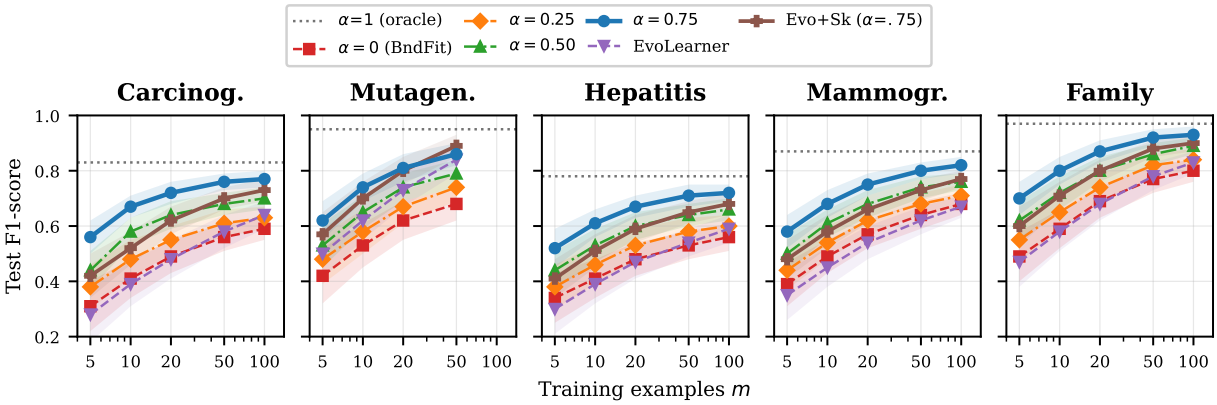

Figure 2: Learning curves: F1 vs. training set size m. Shaded bands: ± 1 std over 10 sketch instantiations $\times$ 5 random draws.

ing is most valuable when the target structure is complex relative to the available data, precisely the regime where the $\|q_T\| \log |\Sigma|$ unrestricted bound is large relative to the sketch bound.

Learner-agnosticism (Q4). Across all five test datasets, Evo + Sketch improves over vanilla EvoLearner in accuracy (Table 2) and convergence speed (Table 3). The two back-ends benefit from sketches through different mechanisms (SAT-based fitting via search space reduction in the formula, evolutionary search via informed initialization and constrained mutation), yet both exhibit the same qualitative pattern: accuracy and speed improve with α. This confirms that sketching is a *framework-level* contribution whose value stems from narrowing the hypothesis space, not from properties of a particular solver.

Scope and validity. The experiments are designed to test the mechanism predicted by the theory: fixing part of the target reduces the completion space and therefore both sample and solver requirements. Because the sketches are derived from known targets rather than authored by experts, the results are controlled evidence for the value of structural information, not a user study. The five benchmarks vary in signature size, target size, and number of examples, but they do not exhaust ontology-engineering practice. Larger controlled scaling studies could separate the effects of signature size, target depth, and hole type more finely.

8 CONCLUSION AND FUTURE WORK

In this paper, we introduced concept sketching for the description logic $\mathcal{ALC}$, letting an expert fix part of a target concept and a back-end learner complete the rest. Beyond the individual results, a signature-governed complexity dichotomy and a per-hole-type sample complexity bound with a matching lower bound, the analysis points to a single underlying quantity. What a sketch shrinks is the completion space, and its size $\log |\mathcal{C}_\psi|$ surfaces in three places at once: it lower-bounds the examples any learner needs (Theorem 5.3), it bounds the enumeration a solver performs (Theorem 4.2), and it tracks the free variables of the SAT encoding (Proposition 6.2). Because $\log |\mathcal{C}_\psi|$ is a property of the sketch and not of any solver, the same structural knowledge pays off statistically and computationally at once, which is why the benefits carry across back-ends as different as SAT-based fitting and evolutionary search. Sketching thus turns expert knowledge into a measurable reduction of the hypothesis space, priced separately for each kind of hole.

Two limits of the present analysis suggest where the framework should grow. First, we take sketches as correct and given, whereas in practice they may be imperfect or absent; this calls for approximate fitting under noisy or unsatisfiable sketches, e.g., via partial weighted Max-SAT, and for interactive tools that help experts author and repair sketches by flagging unsatisfied examples. Second, our guarantees assume ontology-free finite interpretations and a combinatorial back-end; extending them to TBox-mediated reasoning, to neural concept synthesizers through constrained decoding, and settling the one case left open, the complexity of Type-1-only sketches (FPT by h_1 alone, or an inherent $|\Sigma_R|$ dependence), would complete the picture. In each direction the quantity to watch is the same: how much of the hypothesis space the available structure removes.

Acknowledgements

We thank the reviewers for their useful comments. This work was supported by the National Natural Science Foundation of China (No. 62476125), the Noncommunicable Chronic Diseases–National Science and Technology Major Project (No. 2025ZD0550704), the Fundamental Research Funds for the Central Universities (No. 0221-14380025), the "111 Center" (No. B26023), the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM118), and the Nanjing University International Collaboration Initiative.

References

Rajeev Alur, Rastislav Bodík, Garvit Juniwal, Milo M. K. Martin, Mukund Raghothaman, Sanjit A. Seshia, Rishabh Singh, Armando Solar-Lezama, Emina Torlak, and Abhishek Udupa. Syntax-Guided Synthesis. In *Proc. FM-CAD'13*, pages 1–8. IEEE, 2013.

Dana Angluin. Queries and Concept Learning. *Mach. Learn.*, 2(4):319–342, 1988.

Franz Baader and Barbara Morawska. Unification in the description logic EL. *Log. Methods Comput. Sci.*, 6(3), 2010.

Franz Baader and Paliath Narendran. Unification of concept terms in description logics. *J. Symb. Comput.*, 31(3): 277–305, 2001.

Franz Baader, Rafael Peñaloza, and Boontawee Suntisrivaraporn. Pinpointing in the Description Logic $\mathcal{EL}^+$. In *Proc. KI'07*, volume 4667 of *LNCS*, pages 52–67. Springer, 2007.

Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.

Armin Biere, Tobias Faller, Katalin Fazekas, Mathias Fleury, Nils Froleyks, and Florian Pollitt. CaDiCaL 2.0. In *Proc. CAV'24*, volume 14681 of *LNCS*, pages 133–152. Springer, 2024.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis Dimension. *J. ACM*, 36(4):929–965, 1989.

Benjamin Bordais, Daniel Neider, and Rajarshi Roy. Learning Branching-Time Properties in CTL and ATL via Constraint Solving. In *Proc. FM'24*, volume 14933 of *LNCS*, pages 304–323. Springer, 2024.

Benjamin Bordais, Daniel Neider, and Rajarshi Roy. The Complexity of Learning LTL, CTL and ATL Formulas. In *Proc. STACS'25*, volume 327 of *LIPIcs*, pages 19:1–19:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

Cornelius Brand, Robert Ganian, and Kirill Simonov. A Parameterized Theory of PAC Learning. In *Proc. AAAI'23*, pages 6834–6841. AAAI Press, 2023.

Lorenz Bühmann, Jens Lehmann, and Patrick Westphal. DL-Learner - A framework for inductive learning on the Semantic Web. *J. Web Semant.*, 39:15–24, 2016.

William W. Cohen and Haym Hirsh. The Learnability of Description Logics with Equality Constraints. *Mach. Learn.*, 17(2-3):169–199, 1994.

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

Caglar Demir and Axel-Cyrille Ngonga Ngomo. Neuro-Symbolic Class Expression Learning. In *Proc. IJCAI'23*, pages 3624–3632. ijcai.org, 2023.

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

Nicola Fanizzi, Claudia d'Amato, and Floriana Esposito. DL-FOIL Concept Learning in Description Logics. In *Proc. ILP'08*, volume 5194 of *LNCS*, pages 107–121. Springer, 2008.

Michael Frazier and Leonard Pitt. Classic Learning. *Mach. Learn.*, 25(2-3):151–193, 1996.

Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Actively Learning Concepts and Conjunctive Queries under $\mathcal{EL}^r$-Ontologies. In *Proc. IJCAI'21*, pages 1887–1893. ijcai.org, 2021.

Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Frontiers and Exact Learning of $\mathcal{ELI}$ Queries under DL-Lite Ontologies. In *Proc. IJCAI'22*, pages 2627–2633. ijcai.org, 2022.

Maurice Funk, Jean Christoph Jung, and Tom Voellmer. SAT-Based Bounded Fitting for the Description Logic $\mathcal{ALC}$. In *Proc. ISWC'25*, volume 16140 of *LNCS*, pages 42–60. Springer, 2025.

Jean-Raphaël Gaglione, Daniel Neider, Rajarshi Roy, Ufuk Topcu, and Zhe Xu. Learning Linear Temporal Properties from Noisy Data: A MaxSAT-Based Approach. In *Proc. ATVA'21*, volume 12971 of *LNCS*, pages 74–90. Springer, 2021.

Harmender Gahlawat and Meirav Zehavi. Learning Small Decision Trees with Few Outliers: A Parameterized Perspective. In *Proc. AAAI'24*, pages 12100–12108. AAAI Press, 2024.

Robert Ganian, Liana Khazaliya, and Kirill Simonov. Consistency Checking Problems: A Gateway to Parameterized Sample Complexity. In *Proc. IPEC'23*, volume 285 of *LIPIcs*, pages 18:1–18:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

Stefan Heindorf, Lukas Blübaum, Nick Düsterhus, Till Werner, Varun Nandkumar Golani, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. EvoLearner: Learning Description Logics with Evolutionary Algorithms. In *Proc. WWW'22*, pages 818–828. ACM, 2022.

Susmit Jha and Sanjit A. Seshia. A Theory of Formal Synthesis via Inductive Learning. *Acta Informatica*, 54(7): 693–726, 2017.

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Neural Class Expression Synthesis. In *Proc. ESWC'23*, volume 13870 of *LNCS*, pages 209–226. Springer, 2023.

N'Dah Jean Kouagou, Stefan Heindorf, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. ROCES: Robust Class Expression Synthesis in Description Logics via Iterative Sampling. In *Proc. IJCAI'24*, pages 4335–4343. ijcai.org, 2024.

Jens Lehmann and Christoph Haase. Ideal Downward Refinement in the $\mathcal{EL}$ Description Logic. In *Proc. ILP'09*, volume 5989 of *LNCS*, pages 73–87. Springer, 2009.

Jens Lehmann and Pascal Hitzler. Concept learning in description logics using refinement operators. *Mach. Learn.*, 78(1-2):203–250, 2010.

Jens Lehmann, Sören Auer, Lorenz Bühmann, and Sebastian Tramp. Class expression learning for ontology engineering. *J. Web Semant.*, 9(1):71–81, 2011.

Simon Lutz, Daniel Neider, and Rajarshi Roy. Specification Sketching for Linear Temporal Logic. In *Proc. ATVA'23*, volume 14216 of *LNCS*, pages 26–48. Springer, 2023.

Daniel Neider and Ivan Gavran. Learning Linear Temporal Properties. In *Proc. FMCAD'18*, pages 1–10. IEEE, 2018.

Daniel Neider and Rajarshi Roy. What Is Formal Verification Without Specifications? A Survey on Mining LTL Specifications. In *Principles of Verification*, volume 15262 of *LNCS*, pages 109–125. Springer, 2024.

Sebastian Ordyniak and Stefan Szeider. Parameterized Complexity of Small Decision Tree Learning. In *Proc. AAAI'21*, pages 6454–6462. AAAI Press, 2021.

Armando Solar-Lezama. *Program Synthesis by Sketching*. PhD thesis, University of California at Berkeley, 2008.

Armando Solar-Lezama. Program sketching. *Int. J. Softw. Tools Technol. Transf.*, 15(5-6):475–495, 2013.

Armando Solar-Lezama, Liviu Tancau, Rastislav Bodík, Sanjit A. Seshia, and Vijay A. Saraswat. Combinatorial sketching for finite programs. In *Proc. ASPLOS'06*, pages 404–415. ACM, 2006.

Balder ten Cate and Victor Dalmau. Conjunctive Queries: Unique Characterizations and Exact Learnability. *ACM Trans. Database Syst.*, 47(4):14:1–14:41, 2022.

Balder ten Cate, Maurice Funk, Jean Christoph Jung, and Carsten Lutz. SAT-Based PAC Learning of Description Logic Concepts. In *Proc. IJCAI'23*, pages 3347–3355. ijcai.org, 2023.

Leslie G. Valiant. A Theory of the Learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

Steffen van Bergerem, Martin Grohe, and Martin Ritzert. On the Parameterized Complexity of Learning First-Order Logic. In *Proc. PODS'22*, pages 337–346. ACM, 2022.

Steffen van Bergerem, Martin Grohe, and Nina Runde. The Parameterized Complexity of Learning Monadic Second-Order Logic. In *Proc. CSL'25*, volume 326 of *LIPIcs*, pages 8:1–8:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

Patrick Westphal, Lorenz Bühmann, Simon Bin, Hajira Jabeen, and Jens Lehmann. SML-Bench - A benchmarking framework for structured machine learning. *Semantic Web*, 10(2):231–245, 2019.

Changjian Zhang, Parv Kapoor, Ian Dardik, Leyi Cui, Rômulo Meira-Góes, David Garlan, and Eunsuk Kang. Constrained LTL Specification Learning from Examples. In *Proc. ICSE'25*, pages 629–641. IEEE, 2025.

Appendix: Table of Contents

A PROOF OF THEOREM 4.4

We restate and give the full proof.

Theorem 4.4 (W[2]-hardness). *Sketch fitting parameterized by (h, b) is W[2]-hard, even when restricted to Type-0-only sketches with $b = 1$ and $\Sigma_R = \emptyset$.*

Proof. We reduce from HITTING SET, which is W[2]-complete parameterized by solution size k [Downey and Fellows, 2013].

Construction. Given $(U, \mathcal{F}, k)$ with $U = \{u_1, \ldots, u_n\}$ and $\mathcal{F} = \{S_1, \ldots, S_m\}$, we assume without loss of generality that $\mathcal{F} \neq \emptyset$ and each $S_j \neq \emptyset$ (an empty subset makes the instance trivially unsatisfiable, and an empty family trivially satisfiable). Set $\Sigma_C = U$, $\Sigma_R = \emptyset$. The sketch is $\psi = ?^0_1 \sqcup ?^0_2 \sqcup \cdots \sqcup ?^0_k$ with $b = 1$. For each $S_j \in \mathcal{F}$, construct $\mathcal{I}_j$ with $\Delta^{\mathcal{I}_j} = \{a_j, d_j\}$ and $u_i^{\mathcal{I}_j} = \{a_j\}$ if $u_i \in S_j$, $\emptyset$ otherwise. Set $P = \{(\mathcal{I}_j, a_j)\}_{j=1}^m$, $N = \{(\mathcal{I}_j, d_j)\}_{j=1}^m$.

Monotonization. We show any fitting completion can use only concept names. Each hole has $b = 1$, so candidates are $\top$, $\bot$, or $u_i \in \Sigma_C$. If $\sigma(?^0_i) = \bot$, it contributes nothing to the disjunction; since $\mathcal{F} \neq \emptyset$, some $u_\ell \in U$ belongs to some S_j, so replacing $\bot$ by u_ℓ can only enlarge the extension (preserving positive examples) while $d_j \notin u_\ell^{\mathcal{I}_j}$ for all j (preserving negative examples). If $\sigma(?^0_i) = \top$, then $\top^{\mathcal{I}_j} = \{a_j, d_j\}$, violating all negative examples; hence $\top$ cannot appear.

Forward direction. If $H = \{u_{i_1}, \ldots, u_{i_k}\}$ is a hitting set, define $\sigma(?^0_\ell) = u_{i_\ell}$. For each S_j, some $u_{i_\ell} \in S_j$, so $a_j \in u_{i_\ell}^{\mathcal{I}_j} = \{a_j\}$, and the disjunction accepts a_j. Since $u_i^{\mathcal{I}_j} \subseteq \{a_j\}$ for all i, no d_j is accepted.

Reverse direction. If σ is fitting, by monotonization assume $\sigma(?^0_\ell) = u_{i_\ell} \in \Sigma_C$. Set $H = \{u_{i_1}, \ldots, u_{i_k}\}$. For each S_j, acceptance of a_j requires $a_j \in u_{i_\ell}^{\mathcal{I}_j}$ for some ℓ, which implies $u_{i_\ell} \in S_j$.

Parameter preservation. $(h, b) = (k, 1)$; since k is the HITTING SET parameter and b is constant, this is a valid parameterized reduction. □

B PROOF OF THEOREM 6.1

We restate and prove the correctness of the sketch-constrained SAT encoding.

Theorem 6.1 (Correctness). *The formula Φ_ψ is satisfiable if and only if there exists a completion σ of ψ such that $\psi[\sigma]$ fits (P, N).*

Proof. ($\Rightarrow$) Suppose τ is a satisfying assignment of Φ_ψ. We construct a completion σ and show $\psi[\sigma]$ fits (P, N).

For each Type-0 hole $?^0_i$ at position v in the sketch, the free subtree of b nodes rooted at v is encoded with the full set of constructor variables. The one-hot and propagation clauses guarantee that the assignment τ restricted to these b nodes determines a unique $\mathcal{ALC}$ concept $\sigma(?^0_i) \in \mathcal{L}_{\Sigma,b}$: exactly one constructor variable is true at each node, and the tree structure is respected by the propagation clauses.

For each Type-1 hole $?^1_j$ at position v, the encoding restricts the constructor variables to $\{\mu_{v,\neg},\ \mu_{v,\exists r},\ \mu_{v,\forall r} \mid r \in \Sigma_R\} \cup \{\mu_{v,\mathrm{id}}\}$ with a one-hot constraint. The unique true variable determines $\sigma(?^1_j) \in U_\Sigma$. If $\mu_{v,\mathrm{id}}$ is true, the propagation rule $\eta_{v,j,d} \Leftrightarrow \eta_{v_c,j,d}$ ensures that the node acts as the identity function.

For each Type-2 hole $?^2_k$ at position v, exactly one of $\mu_{v,\sqcap}$, $\mu_{v,\sqcup}$ is true, determining $\sigma(?^2_k) \in \{\sqcap, \sqcup\}$.

For each fixed node v with constructor c, the unit clause forces the constructor variable for c to be true and all others false.

The semantics variables $\eta_{v,j,d}$ propagate correctly through the entire tree because: (i) at fixed nodes, the known constructor determines the semantics via the propagation clauses; (ii) at Type-0 holes, the free subtree encoding enforces semantics identically to the unrestricted encoding; (iii) at Type-1 holes, the active constructor determines the semantics (with id handled by the identity propagation rule); (iv) at Type-2 holes, the active binary constructor determines whether the node

computes intersection or union. Therefore, $\eta_{\mathrm{root},j,d}$ is true under τ iff $d \in \psi[\sigma]^{\mathcal{I}_j}$. The fitting clauses then ensure $\psi[\sigma]$ fits (P, N).

($\Leftarrow$) Suppose σ is a completion such that $\psi[\sigma]$ fits (P, N). We construct a satisfying assignment τ for Φ_ψ.

For each fixed node v with constructor c, set the constructor variable for c to true and all others to false. For each Type-0 hole $?_i^0$, assign the constructor variables of the free subtree according to $\sigma(?_i^0)$'s syntax tree (unused nodes set to $\bot$). For each Type-1 hole $?_j^1$, set the variable corresponding to $\sigma(?_j^1)$ to true. For each Type-2 hole $?_k^2$, set $\mu_{v,\sigma(?_k^2)}$ to true.

Set semantics variables by evaluation: for each node v, $\eta_{v,j,d} = 1$ iff $d \in C_v^{\mathcal{I}_j}$, where C_v is the subconcept at v in $\psi[\sigma]$. The propagation clauses are satisfied because they encode $\mathcal{ALC}$ semantics. The fitting clauses are satisfied because $\psi[\sigma]$ fits (P, N). □

C PROOF OF PROPOSITION 6.2

Proof. The sketch encoding tree has $f + h_0 b + h_1 + h_2$ nodes: f fixed, $h_0 b$ from expanding Type-0 holes into subtrees of size b, and $h_1 + h_2$ from Type-1 and Type-2 holes. Each of the f fixed nodes contributes zero free constructor variables (eliminated by unit propagation), saving κ per node. Each Type-0 subtree node uses κ variables (same as unrestricted), contributing $h_0 b\kappa$ in total. Each Type-1 hole uses $2|\Sigma_R|+2$ variables (only unary constructors plus identity), saving $\kappa - (2|\Sigma_R|+2)$. Each Type-2 hole uses 1 variable (the $\sqcap$-vs-$\sqcup$ bit), saving $\kappa-1$. Summing gives the stated reduction. □

D PROOF OF COROLLARY 5.2

Proof. We give the comparison in full and then show that the coefficient 2 on h_1 is tight. The bounded fitting bound is $O(\varepsilon^{-1}(\|q_T\| \log |\Sigma| + \log \delta^{-1}))$; the sketch bound (1) has the terms $h_0 b \log |\Sigma| + h_1 \log(2|\Sigma_R|+2) + h_2$ in place of $\|q_T\| \log |\Sigma|$. We compare these expressions directly.

Since $\Sigma_C \neq \emptyset$, we have $|\Sigma_R| \leq |\Sigma| - 1$, hence $2|\Sigma_R|+2 \leq 2|\Sigma|$. For $|\Sigma| \geq 2$, moreover, $2|\Sigma| \leq |\Sigma|^2$, so

$$h_1 \log(2|\Sigma_R|+2) \;\leq\; h_1 \log(|\Sigma|^2) \;=\; 2h_1 \log |\Sigma|.$$

Using additionally $h_2 \leq h_2 \log |\Sigma|$ (as $\log |\Sigma| \geq 1$ for $|\Sigma| \geq 2$), the sketch terms are bounded by

$$h_0 b \log |\Sigma| + h_1 \log(2|\Sigma_R|+2) + h_2 \;\leq\; (h_0 b + 2h_1 + h_2) \log |\Sigma|.$$

The condition $h_0 b + 2h_1 + h_2 < \|q_T\|$ then gives, since $\log |\Sigma| > 0$,

$$h_0 b \log |\Sigma| + h_1 \log(2|\Sigma_R|+2) + h_2 \;<\; \|q_T\| \log |\Sigma|,$$

so the sketch bound is strictly tighter than the bounded fitting bound.

The coefficient 2 on h_1 cannot be lowered to 1: with $\Sigma_C = \{A\}$, $\Sigma_R = \{r, s, t, u\}$ (so $|\Sigma| = 5$), a sketch with $h_1 = 3$ nested Type-1 holes over $\top$, and target $q_T = \exists r.\exists s.\exists t.\top$ of size $\|q_T\| = 4$, the count condition with coefficient 1 would read $3 < 4$ and be satisfied, yet $3 \log_2 10 \approx 9.97 > 9.29 \approx 4 \log_2 5$, so the sketch expression exceeds the bounded fitting expression. Each Type-1 hole must be priced at two nodes, matching the two syntax tree nodes that a quantified constructor $\exists r.\cdot$ occupies in the target. □

E PROOF OF THEOREM 5.3

Proof. By the standard sample-complexity lower bound for probably approximately correct learning [Blumer et al., 1989], any algorithm that (ε, δ)-learns a concept class of Vapnik-Chervonenkis (VC) dimension d requires $\Omega(\varepsilon^{-1}(d + \log \delta^{-1}))$ examples. It therefore suffices to exhibit a family of sketches for which $\mathrm{VCdim}(\mathcal{C}_\psi) = \Omega(h_0\, b \log |\Sigma| + h_1 \log(2|\Sigma_R|+2) + h_2)$. We construct three independent gadgets, one per hole type, and combine them.

Type-2 gadget. For the k-th Type-2 hole, use the subformula $(A_k \;?_k^2\; B_k)$ with $A_k, B_k \in \Sigma_C$. Take an interpretation $\mathcal{I}_k$ and individual a_k with $a_k \in A_k^{\mathcal{I}_k}$ and $a_k \notin B_k^{\mathcal{I}_k}$. Then the completion $\sqcap$ rejects $(\mathcal{I}_k, a_k)$ while $\sqcup$ accepts it, so the choice at hole k sets the label of $e_k = (\mathcal{I}_k, a_k)$ independently of the other holes. The h_2 examples $\{e_1, \ldots, e_{h_2}\}$ are shattered, giving VC dimension at least h_2.

Table 5: Comparison of DL concept learners. *Sketch compatibility* indicates whether the method's internal representation admits sketch constraints (fixed nodes and typed holes) without fundamental architectural changes.

Method	DL	Mechanism	PAC	Sketch compatibility	Role in this paper
SPELL [ten Cate et al., 2023]	$\mathcal{ELH}^r$	SAT-based bounded fitting	Yes	High (same paradigm)	— (targets $\mathcal{EL}$, not $\mathcal{ALC}$)
SPELL-$\mathcal{ALC}$ [Funk et al., 2025]	$\mathcal{ALC}$	SAT-based bounded fitting	Yes	**Perfect**: fixed nodes → unit clauses	Base model (BoundedFit/SketchFit)
EvoLearner [Heindorf et al., 2022]	$\mathcal{ALCQ}(\mathcal{D})$	Evolutionary (AST repr.)	No	**Good**: sketch-seeded init, hole-only mutation	Second back-end (Evo+Sketch)
CELOE [Bühmann et al., 2016]	$\mathcal{ALCQ}(\mathcal{D})$	Refinement operator	No	Poor: top-down ρ conflicts with fixed subtrees	External reference
Lehmann & Hitzler [Lehmann and Hitzler, 2010]	$\mathcal{ALCQ}$	Refinement operator (theory)	No	Poor (same as CELOE)	— (CELOE represents family)
Lehmann & Haase [Lehmann and Haase, 2009]	$\mathcal{EL}$	Ideal refinement for $\mathcal{EL}$	No	N/A	— (wrong DL: $\mathcal{EL} \subsetneq \mathcal{ALC}$)
DL-FOIL [Fanizzi et al., 2008]	$\mathcal{SHOIQ}(\mathcal{D})$	FOIL-like covering	No	Poor: greedy specialization from $\top$	— (covering incompatible)
DRILL [Demir and Ngomo, 2023]	$\mathcal{ALC}$	Deep RL-guided refinement	No	Difficult: needs state/reward redesign	— (deferred to future work)
NCES [Kouagou et al., 2023]	$\mathcal{ALC}$	Neural seq2seq synthesis	No	Difficult: needs constrained decoding	— (deferred to future work)

Type-1 gadget. A Type-1 hole $?^1(\psi)$ ranges over the $2|\Sigma_R| + 2$ unary constructors $\{\mathrm{id}, \neg\} \cup \{\exists r, \forall r : r \in \Sigma_R\}$. Index the roles $r_1, \ldots, r_{|\Sigma_R|}$ and write each index in binary using $\lceil \log_2 |\Sigma_R| \rceil$ bits. For bit position p, build an example in which the individual a has an r_q-successor satisfying ψ exactly when bit p of q is 1. Then the label vector of $\exists r_q.\psi$ across these positions is the binary code of q, so distinct roles yield distinct label vectors and the $\lceil \log_2 |\Sigma_R| \rceil$ examples are shattered. The constructors id and $\neg$ add two further independently realizable labelings, so a single Type-1 hole shatters $\Omega(\log(2|\Sigma_R|+2))$ points, and h_1 holes contribute $\Omega(h_1 \log(2|\Sigma_R|+2))$.

Type-0 gadget. A Type-0 hole with size bound b may be completed by any monotone conjunction of at most b concept names drawn from Σ_C. The class of such bounded-size monotone conjunctions over $|\Sigma_C|$ names has VC dimension $\Theta(b \log(|\Sigma_C|/b)) = \Theta(b \log |\Sigma|)$ for $b \le |\Sigma_C|^{1-\Omega(1)}$, a classical fact about monotone conjunctions [Blumer et al., 1989]: identifying each example with the set of names holding at its individual, a conjunction $\bigsqcap_{A \in S} A$ accepts an example iff S is contained in that set, and a suitable set of $\Theta(b \log |\Sigma|)$ examples is shattered. Each Type-0 hole thus contributes $\Omega(b \log |\Sigma|)$, and h_0 holes contribute $\Omega(h_0\, b \log |\Sigma|)$.

Combination. Place the $h_0 + h_1 + h_2$ gadgets at disjoint holes of a single sketch, over disjoint example sets, arranging the remaining structure so that on each gadget's examples every other gadget evaluates to a fixed neutral value (for instance, a conjunction of the gadgets, with the non-active gadgets set to $\top$ on those examples). Labelings of the gadgets are then independent, so the union of the shattered sets is shattered, giving

$$\mathrm{VCdim}(\mathcal{C}_\psi) = \Omega(h_0\, b \log |\Sigma| + h_1 \log(2|\Sigma_R|+2) + h_2)\,.$$

Substituting into the lower bound above yields the theorem. The construction exhibits one family of sketches attaining the bound, which certifies that the upper bound of Theorem 5.1 is worst-case optimal; the VC dimension of any particular sketch depends on its structure. □

F DL CONCEPT LEARNER COMPARISON

Table 5 surveys existing DL concept learners along four dimensions: DL expressivity, learning mechanism, PAC guarantee, and sketch compatibility. We briefly justify the selection of baselines and back-ends used in our experiments.

SPELL-$\mathcal{ALC}$ as base model. SPELL-$\mathcal{ALC}$ [Funk et al., 2025] extends the SPELL system [ten Cate et al., 2023] from $\mathcal{ELH}^r$ to $\mathcal{ALC}$. It encodes bounded concept fitting as a propositional SAT instance over a size-bounded syntax tree, where each node carries a constructor variable. Sketch constraints integrate directly: fixed nodes become unit clauses eliminating constructor variables, while holes remain as free variables. The resulting encoding is smaller, directly yielding solver speedups (Table 3). BoundedFit is SPELL-$\mathcal{ALC}$ without sketch constraints ($\alpha = 0$); SketchFit is our extension that adds sketch constraints.

EvoLearner as second back-end. EvoLearner [Heindorf et al., 2022] represents concepts as abstract syntax trees and evolves them via crossover and mutation. Its abstract syntax tree representation admits a natural sketch integration without architectural changes: the initial population is seeded with random sketch completions (replacing biased random walks), and crossover/mutation are restricted to hole positions. This demonstrates that sketch benefits are learner-agnostic.

Table 6: Benchmark datasets from SML-Bench [Westphal et al., 2019] and DL-Learner [Bühmann et al., 2016]. $|E|$: total labeled examples (positive/negative split in parentheses). $\|q_T\|$: target concept size. #Axioms: ABox assertion count (TBox axioms excluded per Remark 2.2).

Dataset	$\|\Sigma_C\|$	$\|\Sigma_R\|$	$\|E\|$	$\|\|q_T\|\|$	#Axioms	**Expressivity**
Carcinog.	142	4	298 (182/116)	10	74 566	$\mathcal{ALC}(\mathcal{D})$
Mutagen.	86	5	84 (42/42)	5	62 066	$\mathcal{AL}(\mathcal{D})$
Hepatitis	14	5	500 (171/329)	7	73 114	$\mathcal{ALE}(\mathcal{D})$
Mammogr.	19	3	961 (516/445)	6	6 808	$\mathcal{AL}(\mathcal{D})$
Family	18	15	174 (82/92)	8	2 040	$\mathcal{ALCH}$

CELOE as external reference. CELOE [Bühmann et al., 2016] is the most widely used refinement-based DL learner. Its top-down refinement strategy (starting from $\top$ and iteratively specializing) is fundamentally incompatible with the fixed-node constraints of sketches, since sketch nodes occur at arbitrary positions in the syntax tree rather than being built incrementally from the root. It serves as an external accuracy reference, not as a sketch integration candidate.

Methods not selected. DRILL [Demir and Ngomo, 2023] uses deep Q-learning to guide refinement-based search. Sketch integration would require redesigning the RL state representation to encode fixed-node constraints and retraining the Q-network, a non-trivial engineering effort. NCES [Kouagou et al., 2023] synthesizes concepts via neural sequence-to-sequence translation. Enforcing sketch structure during autoregressive decoding would require constrained decoding (forcing specific tokens at fixed positions) and per-KB model retraining. Lehmann and Hitzler [2010] provide the theoretical refinement operator foundations that CELOE implements; including both would be redundant. Lehmann and Haase [2009] target $\mathcal{EL}$, which lacks the negation, disjunction, and universal restriction constructors required by our $\mathcal{ALC}$ sketch formalization. DL-FOIL [Fanizzi et al., 2008] uses a FOIL-like greedy covering strategy that builds concepts incrementally from $\top$, making it incompatible with mid-tree fixed constraints. We leave DRILL and NCES integration as future work (Section 8).

G EXPERIMENTAL DETAILS

Sketch generation procedure. Given q_T and retention fraction $\alpha \in (0, 1]$, we uniformly sample $\lceil \alpha \cdot \|q_T\| \rceil$ nodes to fix; the remainder are replaced by holes typed according to their arity in q_T: leaves become Type-0 holes ($b = 1$), unary nodes become Type-1, binary nodes become Type-2. Adjacent unfixed nodes are not merged, preserving a one-to-one correspondence between holes and positions in q_T's syntax tree; this ensures every retention fraction α produces sketches of the same size $\|\psi\| = \|q_T\|$. For BoundedFit ($\alpha = 0$), the entire concept is a single Type-0 hole with $b = \|q_T\|$.

Reproducibility. All solver code, benchmark preprocessing scripts, and evaluation pipelines will be released upon publication. Random sketch generation uses a fixed seed per (dataset, fold, sketch index) triple to ensure full reproducibility.

Hyperparameters. CaDiCaL 1.9.5 in single-threaded mode with default preprocessing (inprocessing, vivification enabled); no proof logging. CELOE uses DL-Learner 1.5 with Pellet 2.3 as OWL reasoner. Key CELOE parameters: noise threshold 30%, max execution time 300 s, max concept length = $\|q_T\| + 3$ (slightly above s to avoid unfairly limiting CELOE), negation and disjunction enabled to match $\mathcal{ALC}$ expressivity; all other parameters at defaults. For SketchFit and BoundedFit, the size bound is $s = \|q_T\|$ for all datasets (see Table 6 for $\|q_T\|$ values); this oracle choice isolates the effect of structural knowledge from the effect of size estimation.

EvoLearner hyperparameters. Both vanilla EvoLearner and Evo + Sketch use population size $N_{\text{pop}} = 100$, maximum $G_{\max} = 200$ generations, tournament size 5, crossover probability 0.9, mutation probability 0.1, and 300 s timeout (same as the SAT solvers). For Evo + Sketch, the initial population consists of N_{pop} independent random completions of the given sketch; for vanilla EvoLearner, the default biased-random-walk initialization of Heindorf et al. [2022] is used. Concept size is bounded by $\|q_T\| + 3$ to match CELOE's setting. All experiments use the ontolearn 0.7 implementation.

Cross-validation protocol. Stratified 10-fold, preserving the positive/negative ratio. We report mean and standard deviation over 10 folds $\times$ 10 sketch instances ($= 100$ evaluations per (dataset, α)). Timeouts are scored as F1 $= 0$ (occurred in 2 of 100 BoundedFit evaluations on Carcinogenesis).