FUZZYCELL: DIFFERENTIABLE CELL ONTOLOGY REASONING FOR GRADED CELL TYPE ANNOTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

How reliable are single-cell foundation models? We audit scGPT, Geneformer, CellFM, and scCello on Tabula Sapiens and discover that one in every five to seven predictions violates a hard biological law encoded in the Cell Ontology (CL): neurons placed in liver, cells simultaneously assigned to mutually exclusive lineages. These errors are invisible to standard metrics, yet they follow from elementary reasoning over CL's 65,000+ logical axioms, which formalize cell type semantics in description logic. The root cause is that every existing model treats annotation as flat classification, discarding the formal knowledge base that defines what a cell type *is*. We propose a paradigm shift: annotation as ontology-constrained entity linking. FUZZYCELL, our framework, retrieves candidate types with a pretrained encoder and re-ranks them with a differentiable fuzzy description logic reasoner built on a new operator, the *Sigmoidal Reichenbach implication*, which provably eliminates the gradient degeneracy that renders standard fuzzy implications untrainable while converging to classical logic in the limit. On Tabula Sapiens, HLCA, and AIDA v2, FUZZYCELL cuts axiom violations by over 80% and lifts Macro-F1 by 4–8 points over the respective backbone, with the largest gains on rare and zero-shot types, as a backbone-agnostic plug-in adding <4 ms per cell. Beyond hard labels, FUZZYCELL outputs graded memberships: a differentiating progenitor is scored $\mu(\text{CLP})=0.42$, $\mu(\text{pro-B})=0.28$, $\mu(\text{immature T})=0.15$, capturing the biological reality that cells live on a continuum, not in boxes.

1 INTRODUCTION

The Human Cell Atlas aims to map every cell type in the human body (A et al., 2017). Single-cell transcriptome foundation models (TFMs) such as scGPT (Cui et al., 2024), Geneformer (Theodoris et al., 2023), and CellFM (Zeng et al., 2025) are its most scalable annotation engine, labeling millions of cells in minutes. Yet these models treat cell types as flat, independent categories, ignoring the Cell Ontology (CL) (Bard et al., 2005; Diehl et al., 2016), the formal knowledge base that defines what a cell type *is*. CL encodes $\sim$2,700 cell types connected by 65,000+ logical axioms across 19,000+ classes (with imports from Uberon and other OBO ontologies) (Tan et al., 2026): every T cell is a lymphocyte (subsumption), every hepatocyte resides in liver tissue (existential restriction), and no cell is simultaneously a neuron and an epithelial cell (disjointness). These axioms are not optional metadata; they are biological laws. A model that violates them is producing annotations that are *biologically impossible*.

We audit four leading TFMs on Tabula Sapiens (The Tabula Sapiens Consortium, 2022), a multi-organ atlas of 483K cells across 24 tissues, and find that 14–25% of predictions violate at least one CL axiom. These are not borderline errors: they include neurons placed in liver and cells assigned to two mutually exclusive lineages. Standard metrics (accuracy, F1) are completely blind to these failures. The only ontology-aware TFM, scCello (Yuan et al., 2024), and a concurrent hierarchical loss (di Montesano et al., 2026) each exploit fragments of CL (graph proximity and the *is_a* hierarchy, respectively) but neither reasons over disjointness or role restrictions. No existing method reasons over the full axiomatic content of the Cell Ontology.

Bridging this gap requires overcoming two barriers. The first is the clash between logic and learning: CL is expressed in the description logic $\mathcal{EL}_\bot$ (Baader et al., 2005; 2017), which admits polynomial-time reasoning (Kazakov et al., 2014) but is Boolean, producing no gradients for neural training.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Fuzzy logic offers a natural relaxation, replacing $\{0, 1\}$ membership with continuous values in $[0, 1]$ (Badreddine et al., 2022; Diligenti et al., 2017), but introduces a second, subtler barrier: *implication bias* (van Krieken et al., 2022). The standard fuzzy implication used to encode subsumption axioms has a gradient that vanishes whenever the consequent membership is high. In biological terms, if the model is already confident that a cell is a lymphocyte, the axiom "every T cell is a lymphocyte" provides zero gradient to refine the T cell membership, even if it is wrong. The optimizer satisfies axioms by inflating all membership degrees rather than learning meaningful type boundaries. This pathology affects every standard residual implication (Reichenbach, Gödel, Goguen), whose gradients vanish on at least half of $[0, 1]^2$ (van Krieken et al., 2022).

We introduce FUZZYCELL (Figure 1), the first framework to integrate differentiable $\mathcal{EL}_\perp$ reasoning into single-cell annotation by solving both barriers at once. FUZZYCELL recasts annotation as *ontology-constrained entity linking*: a pretrained TFM retrieves candidate types, and a differentiable fuzzy reasoner re-ranks them by scoring each candidate's consistency with the full axiom set of CL. The key innovation is the *Sigmoidal Reichenbach implication*, a novel fuzzy operator that provably eliminates gradient degeneracy across the entire $(0, 1)^2$ domain (Proposition 3.1) while converging to classical Boolean logic in the limit (Theorem 3.2).

Beyond eliminating impossible predictions, FUZZYCELL produces a *graded membership vector* $\boldsymbol{\mu}(\mathbf{x}) \in [0, 1]^{|\mathcal{C}(\mathbf{x})|}$ that captures a biological reality hard classifiers erase. Cell differentiation is a continuous process (Trapnell et al., 2014): RNA velocity (Manno et al., 2018) and pseudotime analysis (Wolf et al., 2019) have established that single cells occupy a continuum of states between discrete categories. A common lymphoid progenitor transitioning toward the T cell lineage is simultaneously "part progenitor, part committed T cell," and FUZZYCELL expresses this as $\mu(\text{CLP}) = 0.42$, $\mu(\text{pro-B}) = 0.28$, $\mu(\text{immature T cell}) = 0.15$ while provably respecting CL axioms.

Our contributions are threefold:

1. **Framework.** The first retrieve-then-reason framework for ontology-grounded cell type annotation, recasting annotation as ontology-constrained entity linking. FUZZYCELL is backbone-agnostic: it wraps scGPT, scCello, Geneformer, and CellFM as a plug-in module adding only 0.46M parameters and <4 ms overhead per cell, while producing graded membership vectors that capture transitional cell states invisible to hard classifiers.
2. **Theory.** We introduce the Sigmoidal Reichenbach implication I_σ, the first fuzzy operator to provably eliminate gradient degeneracy across the entire $(0, 1)^2$ domain (Proposition 3.1), resolving an open trainability problem shared by all standard residual implications (Reichenbach, Gödel, Goguen, Łukasiewicz). We further prove that FUZZYCELL's graded annotations converge to valid crisp $\mathcal{EL}_\perp$ models as $s \to \infty$ (Theorem 3.2), guaranteeing semantic soundness.
3. **Experiments.** On Tabula Sapiens (483K cells, 476 types), HLCA (580K cells, 61 types), and AIDA v2 (212K cells, 85 types), FUZZYCELL reduces Axiom Violation Rate by over 80% and improves Macro-F1 by 4–8 percentage points over the respective backbone, with the largest gains on rare and zero-shot types, while matching or exceeding all baselines on common types. Eight qualitative case studies across all three benchmarks validate the reasoning mechanism.

2 PRELIMINARIES

2.1 CELL TYPE ANNOTATION

A single-cell RNA-sequencing experiment produces a dataset $\mathcal{D} = \{(\mathbf{x}_i, c_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^G_{\geq 0}$ is a gene expression vector over G genes and c_i is the tissue of origin. The *cell type annotation* task assigns each cell a type from CL. Current methods output a single hard label; FUZZYCELL additionally produces a *graded membership vector* $\boldsymbol{\mu}(\mathbf{x}) \in [0, 1]^{|\mathcal{C}(\mathbf{x})|}$ over the candidate set $\mathcal{C}(\mathbf{x})$ (defined in Section 3.1), motivated by the biological reality that differentiating cells occupy intermediate states (Trapnell et al., 2014).

2.2 THE CELL ONTOLOGY AS AN $\mathcal{EL}_\perp$ KNOWLEDGE BASE

The Cell Ontology is not a mere list of names; it is a logic-based knowledge base maintained by the Open Biological and Biomedical Ontologies (OBO) Foundry (Smith et al., 2007) and expressed in

the Web Ontology Language (OWL) (Bechhofer, 2018) under the $\mathcal{EL}$ profile (Hoehndorf et al., 2011). Its formal underpinning is the description logic $\mathcal{EL}_\bot$ (Baader et al., 2005), which we now define.

Syntax. An $\mathcal{EL}_\bot$ *signature* consists of a finite set of *concept names* $\mathsf{N_C}$ (representing cell types) and *role names* $\mathsf{N_R}$ (representing biological relationships such as *part_of* or *develops_from*). *Concept expressions* are formed by the grammar

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

where $A \in \mathsf{N_C}$ is a concept name, also called an *atomic concept* to distinguish it from compound expressions built by the remaining constructors (e.g., *hepatocyte* is atomic, while *hepatocyte* $\sqcap$ $\exists$*part_of.liver* is compound). The binary operator $\sqcap$ denotes *conjunction* ("cells that are both C and D"), $\exists r.C$ is an *existential restriction* ("cells that stand in relation r to some cell of type C") for role $r \in \mathsf{N_R}$, $\top$ is the universal concept containing every cell, and $\bot$ is the empty concept. A *TBox* $\mathcal{T}$ is a finite set of *general concept inclusions* (GCIs) of the form $C \sqsubseteq D$, read "every C is a D." For example, *hepatocyte* $\sqsubseteq$ *epithelial cell* states that every hepatocyte is an epithelial cell, and *hepatocyte* $\sqsubseteq \exists$*part_of.liver* states that every hepatocyte resides in some liver tissue.

Semantics. An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty domain $\Delta^{\mathcal{I}}$ (intuitively, the set of all individual cells) and a mapping $\cdot^{\mathcal{I}}$ that assigns to each concept name $A \in \mathsf{N_C}$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ (the cells of type A) and to each role $r \in \mathsf{N_R}$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. Compound concepts are interpreted inductively:

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}, \quad (\exists r.C)^{\mathcal{I}} = \{d \in \Delta^{\mathcal{I}} \mid \exists e \in C^{\mathcal{I}} : (d, e) \in r^{\mathcal{I}}\}, \quad \top^{\mathcal{I}} = \Delta^{\mathcal{I}}, \quad \bot^{\mathcal{I}} = \varnothing.$$

An interpretation $\mathcal{I}$ *satisfies* a GCI $C \sqsubseteq D$, written $\mathcal{I} \models C \sqsubseteq D$, when $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, i.e., every cell in C is also in D. Disjointness axioms take the form $C \sqcap D \sqsubseteq \bot$, meaning no cell can simultaneously belong to both C and D. $\mathcal{I}$ is a *model* of a TBox $\mathcal{T}$ if it satisfies every GCI in $\mathcal{T}$. Subsumption reasoning in $\mathcal{EL}_\bot$ is tractable: deciding whether $\mathcal{T} \models C \sqsubseteq D$ is polynomial in the number of axioms in $\mathcal{T}$ (Baader et al., 2005). CL defines $|\mathsf{N_C}| \approx 2{,}700$ cell types and $|\mathsf{N_R}| = 15$ roles (*part_of*, *develops_from*, etc.). Its TBox $\mathcal{T}_{\mathsf{CL}}$ contains 65,825 logical axioms across 19,151 classes, including subsumption, existential restriction, and disjointness axioms spanning both cell types and imported anatomical, functional, and lineage concepts from Uberon and other OBO ontologies (Tan et al., 2026). Every cell in the CELLxGENE atlas (Abdulla et al., 2025) (>93M cells) carries a CL identifier, making $\mathcal{T}_{\mathsf{CL}}$ the de facto standard for cell type semantics.

2.3 Fuzzy Semantics for $\mathcal{EL}_\bot$

In the crisp semantics above, a cell either belongs to a concept or does not: its membership degree is 1 or 0. This binary view conflicts with the biology of differentiating cells, which often reside in transitional states between established categories (Manno et al., 2018). Fuzzy logic offers a principled relaxation by allowing membership degrees in $[0, 1]$, enabling integration of ontological reasoning into gradient-based learning (van Krieken et al., 2022).

Definition 2.1 (Fuzzy interpretation)**.** A *fuzzy interpretation* $\mathcal{I}_f = (\Delta^{\mathcal{I}_f}, \cdot^{\mathcal{I}_f})$ assigns to each concept name $A \in \mathsf{N_C}$ a membership function $A^{\mathcal{I}_f} : \Delta^{\mathcal{I}_f} \to [0, 1]$ and to each role $r \in \mathsf{N_R}$ a function $r^{\mathcal{I}_f} : \Delta^{\mathcal{I}_f} \times \Delta^{\mathcal{I}_f} \to [0, 1]$. Compound concepts are interpreted using a *t-norm* T, a commutative, associative, monotone binary operation on $[0, 1]$ satisfying $T(a, 1) = a$, and a supremum:

$$(C \sqcap D)^{\mathcal{I}_f}(d) = T\big(C^{\mathcal{I}_f}(d),\, D^{\mathcal{I}_f}(d)\big), \tag{1}$$

$$(\exists r.C)^{\mathcal{I}_f}(d) = \sup_{e \in \Delta^{\mathcal{I}_f}} T\big(r^{\mathcal{I}_f}(d, e),\, C^{\mathcal{I}_f}(e)\big), \quad \top^{\mathcal{I}_f}(d) = 1, \quad \bot^{\mathcal{I}_f}(d) = 0.$$

Intuitively, Equation (1) says that a cell's membership in the conjunction $C \sqcap D$ is determined by combining its memberships in C and D via the t-norm. We adopt the *product t-norm* $T_P(a, b) = a \cdot b$, following the analysis of van Krieken et al. (2022), who show it is the only t-norm whose symmetric configuration yields consistent improvements in differentiable learning.

Fuzzy satisfaction and the implication bias problem. In crisp semantics, a GCI $C \sqsubseteq D$ holds when $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, meaning every cell in C is also in D. The fuzzy counterpart replaces set inclusion

with a *fuzzy implication* I applied to each cell's membership degrees, and aggregates the results over a mini-batch $\mathcal{B} \subseteq \Delta^{\mathcal{I}_f}$ using log-product aggregation (van Krieken et al., 2022):

$$\mathrm{sat}_{\mathcal{I}_f}(C \sqsubseteq D) \;=\; \sum_{d \in \mathcal{B}} \log I\big(C^{\mathcal{I}_f}(d),\; D^{\mathcal{I}_f}(d)\big).$$

When every implication evaluates to 1 (fully satisfied), the sum reaches its maximum of 0; lower values indicate axiom violations. This satisfaction score will form the basis of the ontology consistency loss $\mathcal{L}_{\text{onto}}$ in Section 3.3. The standard choice for I is the *Reichenbach implication* (van Krieken et al., 2022):

$$I_R(a, b) \;=\; 1 - a + a \cdot b,$$

whose partial derivatives are $\frac{\partial I_R}{\partial a} = b - 1$ and $\frac{\partial I_R}{\partial b} = a$. When b is close to 1, i.e., when concept D already has high membership, the gradient $\frac{\partial I_R}{\partial a}$ vanishes regardless of a. To give a concrete example: for the axiom "$T\,cell \sqsubseteq lymphocyte$" we have $a = T\,cell^{\mathcal{I}_f}(d)$ and $b = lymphocyte^{\mathcal{I}_f}(d)$. If the model already assigns a cell a high lymphocyte membership ($b \approx 1$), then $\frac{\partial I_R}{\partial a} = b - 1 \approx 0$: the axiom provides essentially zero gradient to the T cell membership a, even if a is incorrectly low. This *implication bias* (van Krieken et al., 2020) incentivizes the optimizer to trivially satisfy all subsumption axioms by inflating every membership degree toward 1, rather than learning meaningful type boundaries. The problem extends to all residual implications (Gödel, Goguen), whose gradients vanish on $\geq 50\%$ of $[0, 1]^2$ (van Krieken et al., 2022). Our solution is a *sigmoidal* variant that eliminates this pathology (Section 3.2).

3 The FuzzyCell Framework

FuzzyCell recasts cell type annotation as a two-stage pipeline (Figure 1): a neural retriever narrows the search to a small set of plausible types, and a differentiable fuzzy reasoner reranks them by their consistency with the Cell Ontology. The entire pipeline is end-to-end differentiable, allowing ontological constraints to shape the encoder's representations during training.

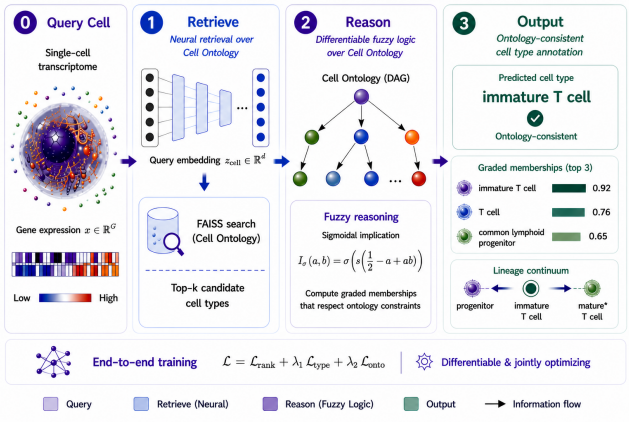

Figure 1: **Overview of FuzzyCell. Input:** A query cell is characterized by its gene expression vector and tissue context. **Stage 1 (Retrieve):** Any pretrained TFM backbone encodes the cell into a dense vector; FAISS retrieves top-k candidate types from the Cell Ontology. **Stage 2 (Reason):** Dual projection maps cell and candidate embeddings into a shared logic space. A differentiable fuzzy $\mathcal{EL}_\perp$ reasoner then scores each candidate's consistency with CL axioms (subsumption, disjointness, existential restrictions) using the Sigmoidal Reichenbach implication I_σ. The output is a graded membership vector $\boldsymbol{\mu}(\mathbf{x}) \in [0, 1]^{|\mathcal{C}(\mathbf{x})|}$, where tissue context constrains the axiom evaluation.

3.1 Candidate Retrieval

Given a query cell $\mathbf{x} \in \mathbb{R}^G_{\geq 0}$, a pretrained TFM backbone f produces a cell embedding $\mathbf{z}_{\text{cell}} = f(\mathbf{x}) \in \mathbb{R}^d$. For each concept name $C \in \mathsf{N_C}$ of CL, a biomedical language model g_ϕ (PubMedBERT; Gu et al. 2022) encodes the concept's preferred name into a type embedding $\mathbf{e}_C = g_\phi(\text{name}(C)) \in \mathbb{R}^{d'}$. The type embeddings are indexed offline using FAISS (Johnson et al., 2021), and for each query cell, the top-k candidates are retrieved by cosine similarity:

$$\mathcal{C}(\mathbf{x}) \;=\; \text{top-k}_{C \in \mathsf{N_C}} \; \frac{\mathbf{z}_{\text{cell}}^\top \mathbf{e}_C}{\|\mathbf{z}_{\text{cell}}\| \cdot \|\mathbf{e}_C\|}.$$

Because FuzzyCell treats the backbone f as a black box, any TFM can be plugged in without architectural modification.

3.2 ONTOLOGICAL CONSISTENCY SCORING

The core of FUZZYCELL is a differentiable module that evaluates each candidate's logical consistency with CL. This module instantiates the abstract fuzzy semantics of Section 2.3 as a concrete neural computation.

Dual projection into logic space. The retrieval space, optimized for nearest-neighbor search, may not align well with the geometric structure needed for logical reasoning. We therefore introduce two learnable linear projections $\pi_{\text{cell}} \colon \mathbb{R}^d \to \mathbb{R}^{d_L}$ and $\pi_{\text{type}} \colon \mathbb{R}^{d'} \to \mathbb{R}^{d_L}$ that map the cell and type embeddings into a shared *logic space*:

$$\mathbf{z}'_{\text{cell}} = \pi_{\text{cell}}(\mathbf{z}_{\text{cell}}), \qquad \mathbf{a}'_\tau = \pi_{\text{type}}(\mathbf{e}_\tau) \quad \text{for each semantic type } \tau \in \Gamma,$$

where $\Gamma \subseteq \mathsf{N_C}$ is a finite set of semantic type concepts used for disambiguation (e.g., *epithelial cell*, *lymphocyte*, *myeloid cell*). The base type embeddings $\{\mathbf{e}_\tau\}$ are frozen; only the projections π_{cell} and π_{type} are learned.

Context-aware type inference. This component gives a concrete neural form to the abstract fuzzy membership function $A^{\mathcal{I}_f}(d) \in [0,1]$ of Definition 2.1. For the query cell (the "domain element" d in the fuzzy interpretation), we compute the degree to which it belongs to each semantic type τ as:

$$\tau^{\mathcal{I}_f}(\mathbf{x}) \;=\; \sigma\Big(\frac{\mathbf{z}'^{\top}_{\text{cell}}\mathbf{a}'_\tau}{\theta}\Big), \tag{2}$$

where σ is the sigmoid and $\theta > 0$ is a learnable temperature controlling the sharpness of the membership distribution. Intuitively, a cell whose projected embedding is close to the projected embedding of "T cell" receives a high T cell membership degree. For candidate concepts, type membership is not inferred but derived from the TBox: $\tau^{\mathcal{I}_f}(C_j) = \mathbb{1}[\mathcal{T}_{\mathsf{CL}} \models C_j \sqsubseteq \tau]$, precomputed offline by the ELK reasoner (Kazakov et al., 2014). This asymmetry localizes all uncertainty to the query cell, while treating the ontology as authoritative ground truth.

The Sigmoidal Reichenbach implication. With membership degrees in hand, we can score type consistency. The natural tool is the Reichenbach implication $I_R(a,b) = 1 - a + ab$ introduced in Section 2.3. However, as discussed there, I_R suffers from implication bias. We resolve this by wrapping I_R in a sigmoid centered at $\frac{1}{2}$:

$$I_\sigma(a,b) \;=\; \sigma\big(s \cdot (I_R(a,b) - \tfrac{1}{2})\big) \;=\; \sigma\big(s \cdot (\tfrac{1}{2} - a + ab)\big),$$

where $s > 0$ controls the steepness. The centering at $\frac{1}{2}$ places the sigmoid's steepest region at the decision boundary of the Reichenbach implication, so that the gradient is concentrated precisely where disambiguation matters most: near-boundary cases where the antecedent is strong but the consequent is ambiguous. As $s \to \infty$, I_σ converges pointwise to a step function at $I_R = \frac{1}{2}$, recovering crisp logical behavior. We prove that this gradient is non-degenerate across the entire $(0,1)^2$ domain (Proposition 3.1 in Section 3.4).

Consistency scoring and score fusion. For each candidate $C_j \in \mathcal{C}(\mathbf{x})$, we evaluate the consistency between the cell's inferred type memberships and the candidate's known type memberships across all semantic types:

$$s_{\text{onto}}(\mathbf{x}, C_j) \;=\; \prod_{\tau \in \Gamma} I_\sigma\big(\tau^{\mathcal{I}_f}(\mathbf{x}),\; \tau^{\mathcal{I}_f}(C_j)\big).$$

This product-aggregation captures the logical intuition that *all* type constraints must be satisfied simultaneously: a candidate that violates even one disjointness axiom receives a near-zero score. In practice, we compute in log-space for numerical stability: $\log s_{\text{onto}} = \sum_\tau \log I_\sigma(\cdot)$. The ontological score is combined with a normalized neural similarity to produce the final ranking score:

$$s_{\text{final}}(\mathbf{x}, C_j) \;=\; \alpha \cdot \tilde{s}_{\text{neural}}(\mathbf{x}, C_j) \;+\; (1-\alpha) \cdot s_{\text{onto}}(\mathbf{x}, C_j),$$

where $\tilde{s}_{\text{neural}} = \frac{1}{2}(\cos(\mathbf{z}'_{\text{cell}}, \mathbf{e}'_{C_j}) + 1) \in [0,1]$ is the rescaled cosine similarity in the logic space, with $\mathbf{e}'_{C_j} = \pi_{\text{type}}(\mathbf{e}_{C_j})$ the projected candidate embedding, and $\alpha \in [0,1]$ balances the two signals. The predicted cell type is $e^* = \arg\max_{C_j \in \mathcal{C}(\mathbf{x})} s_{\text{final}}(\mathbf{x}, C_j)$, and the graded membership vector $\boldsymbol{\mu}(\mathbf{x})$ is the vector of s_{final} values, normalized to sum to one over the candidate set.

3.3 END-TO-END TRAINING

FUZZYCELL is trained with a multi-task objective combining three losses. A *margin-based ranking loss* $\mathcal{L}_{\text{rank}}$ encourages the correct type e^+ to score above in-batch negatives and hard negatives retrieved from the FAISS index (margin γ=0.2). A *type prediction loss* $\mathcal{L}_{\text{type}}$ supervises the context-aware type inference (Equation (2)) via binary cross-entropy against gold labels $y_\tau = \mathbb{1}[\mathcal{T}_{\mathsf{CL}} \models e^+ \sqsubseteq \tau]$, computed by ELK with transitive subsumption. An *ontology consistency loss* directly maximizes fuzzy TBox satisfaction:

$$\mathcal{L}_{\text{onto}} \;=\; -\sum_{(C \sqsubseteq D) \in \mathcal{T}_{\text{batch}}} \log I_\sigma\big(C^{\mathcal{I}_f}(\mathbf{x}),\; D^{\mathcal{I}_f}(\mathbf{x})\big),$$

where $\mathcal{T}_{\text{batch}}$ contains the TBox axioms involving concepts in the current mini-batch. The final objective is $\mathcal{L} = \mathcal{L}_{\text{rank}} + \lambda_1 \mathcal{L}_{\text{type}} + \lambda_2 \mathcal{L}_{\text{onto}}$, where $\lambda_1, \lambda_2 > 0$. Because I_σ is differentiable everywhere on $(0,1)^2$, ontological violations propagate gradients through the projection heads to the TFM backbone, allowing the encoder to learn representations that are both discriminative and ontologically coherent. Full training details (optimizer, schedule, negative sampling, data augmentation) appear in Section D.4.

3.4 THEORETICAL GUARANTEES

The Sigmoidal Reichenbach implication comes with two formal guarantees (full proofs in Section A).

Proposition 3.1 (Gradient non-degeneracy). *Let $I_\sigma(a,b) = \sigma\big(s \cdot (I_R(a,b) - \frac{1}{2})\big)$ with $s > 0$. Then $\|\nabla I_\sigma(a,b)\| > 0$ for all $(a,b) \in (0,1)^2$. In contrast, the Goguen implication $I_{GG}(a,b) = \min(1,\, b/a)$ has $\nabla I_{GG} = \mathbf{0}$ for all $a \le b$ ($\geq$50% of $[0,1]^2$), and the standard Reichenbach I_R has $\partial I_R/\partial a = b - 1$, which vanishes on the entire line $b = 1$.*

The practical consequence is that when training with I_σ, every axiom in the TBox produces a non-zero gradient signal regardless of the current membership values, whereas Goguen-based and Reichenbach-based systems lose signal as membership values approach their boundary values.

Theorem 3.2 (Semantic soundness). *The fuzzy semantics induced by I_σ is a conservative extension of classical $\mathcal{EL}_\bot$: (i) on crisp inputs $a, b \in \{0,1\}$, $\lim_{s\to\infty} I_\sigma(a,b) = \mathbb{1}[a \le b]$, recovering the classical Boolean implication; (ii) if $\mathcal{T}_{\mathsf{CL}} \models C \sqsubseteq D$, then for any crisp interpretation $\mathcal{I}$, $\lim_{s\to\infty} \inf_{d \in \Delta^{\mathcal{I}}} I_\sigma(C^{\mathcal{I}}(d), D^{\mathcal{I}}(d)) = 1$.*

Together, these results guarantee that FUZZYCELL's graded annotations are semantically grounded: they recover exact classical reasoning in the limit, while at finite s the sigmoid provides a smooth interpolation that enables gradient-based learning. In our experiments, $s \in [5, 20]$ offers an effective balance between optimization stability and logical sharpness (Section B.5).

4 EXPERIMENTS

We evaluate FUZZYCELL along four axes: (1) does it reduce ontology violations? (2) does it improve annotation accuracy, especially on rare and zero-shot types? (3) is it truly backbone-agnostic? and (4) which components matter most?

4.1 SETUP

Datasets. **Tabula Sapiens** (The Tabula Sapiens Consortium, 2022) (483K cells, 24 tissues, 476 types), **HLCA** (Sikkema et al., 2023) (580K cells, lung, 61 types), and **AIDA v2** (Yuan et al., 2024) (212K cells, 85 types, cross-dataset generalization). Cell types are partitioned into 60%/20%/20% (train/validation/zero-shot) splits, stratified by lineage; cells within train types are further split 80%/20%. We report mean $\pm$ std over 5 independent runs with different random seeds.

Baselines. *Classical*: SingleR (Aran et al., 2019), scANVI (Xu et al., 2021), CellAssign (Zhang et al., 2019). *TFM (flat)*: scGPT (Cui et al., 2024), Geneformer (Theodoris et al., 2023), CellFM (Zeng et al., 2025). *TFM (ontology-aware)*: scCello (Yuan et al., 2024), HierCE (di Montesano et al., 2026). All baselines use official checkpoints; TFMs are fine-tuned on our training split with the same preprocessing (Section D.6). We evaluate FUZZYCELL on top of each TFM backbone.

Table 1: **Main results on Tabula Sapiens** (483K cells, 476 types, 20% zero-shot types). Mean ± std over 5 independent runs. FUZZYCELL improves every backbone on all metrics ($p < 0.01$, Wilcoxon signed-rank test with Bonferroni correction). Best result per metric in **bold**; best baseline underlined.

Category	Method	Acc. ↑	Macro-F1 ↑	W-F1 ↑	AVR (%) ↓	HD-F1 ↑
Classical	SingleR	78.2 ±0.3	41.3 ±0.8	76.8 ±0.4	22.1 ±1.2	58.4 ±0.6
	scANVI	80.5 ±0.4	44.7 ±0.9	79.1 ±0.5	19.8 ±1.0	61.2 ±0.7
	CellAssign	76.9 ±0.5	38.6 ±1.1	75.2 ±0.6	24.5 ±1.4	55.7 ±0.9
TFM (flat)	scGPT	87.3 ±0.2	52.8 ±0.6	86.1 ±0.3	18.4 ±0.8	69.5 ±0.5
	Geneformer	86.1 ±0.3	50.2 ±0.7	85.0 ±0.3	20.7 ±0.9	67.8 ±0.6
	CellFM	85.8 ±0.3	49.5 ±0.8	84.6 ±0.4	21.3 ±1.0	66.9 ±0.6
TFM (onto-aware)	scCello	88.6 ±0.2	55.1 ±0.5	87.4 ±0.2	14.2 ±0.7	72.3 ±0.4
	HierCE	88.1 ±0.2	54.3 ±0.6	87.0 ±0.3	16.8 ±0.8	71.0 ±0.5
	FUZZYCELL +scGPT	**89.7 ±0.2**	**60.4 ±0.4**	**88.9 ±0.2**	**2.8 ±0.3**	**78.6 ±0.3**
	FUZZYCELL +Geneformer	88.9 ±0.2	58.1 ±0.5	88.0 ±0.3	3.5 ±0.4	76.2 ±0.4
	FUZZYCELL +CellFM	88.5 ±0.3	57.2 ±0.5	87.5 ±0.3	3.9 ±0.4	75.4 ±0.4
	FUZZYCELL +scCello	89.4 ±0.2	59.8 ±0.4	88.6 ±0.2	2.4 ±0.3	77.9 ±0.3

Preprocessing and metrics. We normalize to 10,000 counts per cell, apply $\log(1+x)$, and select the top 2,000 HVGs (Wolf et al., 2018); TFM backbones using their own gene vocabularies follow their official protocols. We report Accuracy, Macro-F1, and Weighted-F1 alongside two ontology-aware metrics (formal definitions in Section C). *Axiom Violation Rate* (AVR) is the fraction of predictions that violate at least one CL disjointness or role-restriction axiom. *Hierarchy-Distance-Weighted F1* (HD-F1) penalizes misclassifications proportionally to their distance in the CL hierarchy, so confusing a T cell with a B cell (siblings) costs less than confusing it with a neuron.

Implementation details. We set k=64, d_L=256, $|\Gamma|$=128, s=10, α=0.6, λ_1=1.0, λ_2=0.5 (selected on the validation set; see Section D.5). Training uses AdamW (Loshchilov & Hutter, 2019) (lr 5×10^{-5}, batch 256) with early stopping on validation Macro-F1, converging in ∼12 epochs (∼4 h on a single A100 GPU). Full details in Section D.4.

4.2 MAIN RESULTS

Table 1 shows results on Tabula Sapiens. FUZZYCELL reduces AVR from 14–25% (baselines) to 2–4%, an over 80% relative reduction, and improves Macro-F1 by 4–8 points over the respective backbone, with the best variant (FUZZYCELL +scGPT) exceeding scCello by 5.3 points. Results on HLCA and AIDA v2 follow the same pattern (Tables 3–4 in the Appendix).

4.3 ANALYSIS

Violation analysis and rare types. Figure 2a decomposes axiom violations into three categories. Flat TFMs violate disjointness axioms at rates of 8–10%; scCello reduces disjointness to 5.8% via graph proximity but cannot eliminate violations because its undirected graph discards directional subsumption. FUZZYCELL reduces all three categories to $\leq$1.2%, with the largest gains on disjointness, the axiom type that no prior method explicitly handles. Figure 2b reveals where these gains matter most: on rare types (<50 cells), FUZZYCELL +scGPT achieves F1 of 0.22 on the rarest types, compared to 0.05 for scGPT alone, a 4× improvement. The mechanism is intuitive: when training data is insufficient to learn a reliable classifier, ontological axioms provide complementary evidence. On types entirely absent from training (zero-shot), the advantage is even starker: FUZZYCELL +scGPT achieves Macro-F1 of 38.5, compared to 22.8 for scGPT and 26.3 for scCello, while reducing AVR from 29.6% to 4.8% (full zero-shot results

Table 2: **Ablation** (FUZZYCELL +scGPT, Tabula Sapiens; mean ± std, 5 independent runs). Cross-dataset ablation in Section B.5.

Variant	M-F1 ↑	AVR ↓
FUZZYCELL +scGPT (full)	**60.4 ±0.4**	**2.8 ±0.3**
w/o $\mathcal{L}_{\text{onto}}$	58.1 ±0.5	6.2 ±0.5
w/o $\mathcal{L}_{\text{type}}$	56.9 ±0.6	3.5 ±0.4
w/o dual projection	57.3 ±0.5	4.1 ±0.4
w/o I_σ (use I_R)	55.8 ±0.7	8.9 ±0.6
w/o tissue context	58.7 ±0.4	4.8 ±0.4

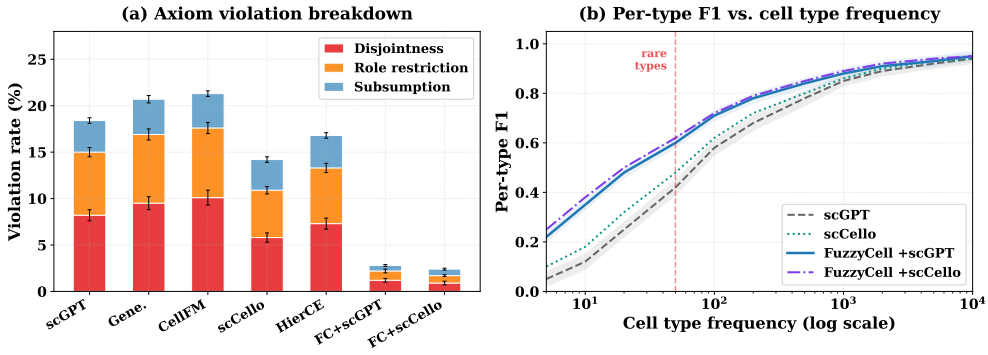

Figure 2: **(a) Axiom violation breakdown on Tabula Sapiens** (error bars: ± 1 std over 5 independent runs; stacked). Flat TFMs violate 18–21% of axioms total. FUZZYCELL nearly eliminates all three violation types by reasoning over the full $\mathcal{EL}_\perp$ axiom set. **(b) Per-type F1 vs. cell type frequency** (shaded bands: ± 1 std). FUZZYCELL yields the largest gains on rare types (<50 cells); on common types, all methods converge.

in Table 6, Section B.3). On common types (>1000 cells), all methods converge, confirming no accuracy trade-off.

Ablation and overhead. Table 2 isolates each component's contribution. Replacing I_σ with the standard Reichenbach I_R causes the largest degradation: Macro-F1 drops by 4.6 points and AVR triples to 8.9%, confirming that implication bias is a practical, not merely theoretical, obstacle. Removing $\mathcal{L}_{\text{onto}}$ doubles AVR while barely affecting Macro-F1, showing that the ontology loss specifically targets violations. FUZZYCELL is robust to hyperparameter choices: $s \in [5, 20]$ and $\alpha \approx 0.6$ are broadly effective (Figure 4 in Section B.5). At inference time, FUZZYCELL's reasoning module adds 3.7–3.9 ms per cell (8–13% overhead relative to the TFM backbone), dominated by the dual projection and type inference (Table 10 in Section B.5).

Qualitative case studies. Figure 3 illustrates three capabilities of FUZZYCELL on individual cells. In **(a)**, scGPT assigns a liver cell to *glutamatergic neuron* (confidence 0.72), violating both a disjointness axiom (*neuron* $\sqcap$ *epithelial cell* $\sqsubseteq \perp$) and a role restriction (*neuron* $\sqsubseteq \exists$*part_of.nervous tissue*, but the cell is from liver). FUZZYCELL scores neuron at 0.003 and correctly predicts *hepatocyte* (0.87). In **(b)**, both scGPT and scCello rank *type I pneumocyte* above *type II* for a lung cell. FUZZYCELL exploits the existential restriction *type II pneumocyte* $\sqsubseteq \exists$*has_part.lamellar body*, consistent with the cell's secretory signature, and flips the ranking. In **(c)**, scGPT assigns the hard label *pro-B cell* to a bone marrow common lymphoid progenitor (CLP). FUZZYCELL instead produces a graded membership vector (μ(CLP)=0.42, μ(pro-B)=0.28, μ(immature T)=0.15), correctly identifying a transitional cell with partial commitment to both B and T cell lineages. Section B.4 provides five additional case studies (HLCA, AIDA v2), and a failure case analysis.

5 RELATED WORK

Cell type annotation and foundation models. TFMs encode cells via masked gene prediction (Cui et al., 2024; Theodoris et al., 2023; Zeng et al., 2025) or contrastive learning (Yuan et al., 2024), but recent benchmarks show they do not consistently outperform simpler methods on zero-shot types (Kedzierska et al., 2025; Boiarsky et al., 2024). Several methods exploit the Cell Ontology's *is_a* hierarchy for annotation: OnClass (Wang et al., 2021) projects cells and CL terms into a shared embedding space with random-walk-based score propagation for zero-shot prediction; CellO (Bernstein et al., 2021) trains a cascade of per-node SVMs along the CL DAG; and SOCAM (Magre et al., 2026) enforces parent–child probability consistency via post-hoc propagation. scCello (Yuan et al., 2024) uses Personalized PageRank on an undirected CL graph, and HierCE (di Montesano et al., 2026) encodes the *is_a* hierarchy, but all these methods treat CL as a graph rather than a logical theory: none reasons over disjointness or role axioms, and all output hard labels. FUZZYCELL reasons over

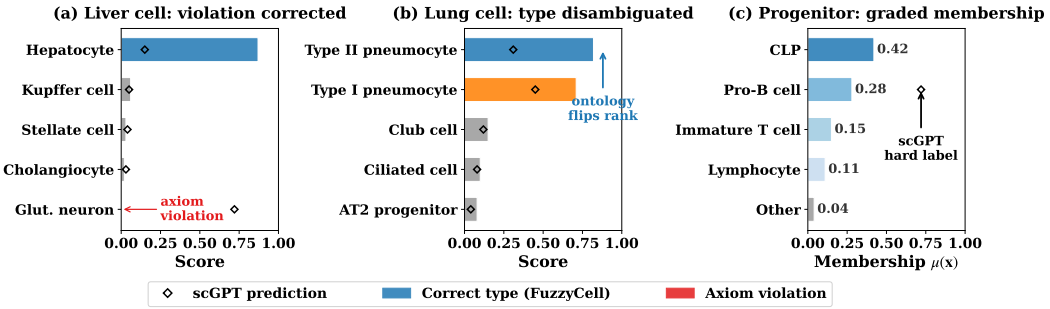

Figure 3: **Qualitative case studies** on Tabula Sapiens. ◇: scGPT prediction; colored bars: FUZZYCELL scores. **(a)** A disjointness violation is corrected: neuron in liver is biologically impossible. **(b)** Ontological reasoning disambiguates two morphologically similar types. **(c)** Graded membership captures the transitional identity of a differentiating progenitor.

the full $\mathcal{EL}_\bot$ axiom set of CL, producing graded annotations that are provably consistent with the ontology; a detailed comparison appears in Section E.

Differentiable fuzzy logic and neuro-symbolic reasoning. Differentiable fuzzy logics (Badreddine et al., 2022; Diligenti et al., 2017; Xu et al., 2018) enable gradient-based optimization of logical constraints. van Krieken et al. (2022) identify the implication bias problem; He et al. (2024) propose the RILL loss as an alternative solution. Recent advances in scalable neuro-symbolic systems (Naik et al., 2025; Choi et al., 2025; Maene et al., 2025) and the hardness results of Maene et al. (2024) further motivate fuzzy (rather than probabilistic) semantics. Wu & Zhao (2025) develop EmALC for knowledge base completion, and FALCON (Tang et al., 2022) approximates semantic entailment over full $\mathcal{ALC}$ ontologies via fuzzy membership functions. In the biological domain, BioFuzzNet (Le Gac et al., 2022) applies propositional fuzzy logic with autograd-based optimization to model signal transduction networks at single-cell scale, though without ontological reasoning. FUZZYCELL is the first to bring differentiable $\mathcal{EL}_\bot$ reasoning to cell type annotation.

Ontology embeddings and biomedical entity linking. EL Embeddings (Kulmanov et al., 2019), BoxEL (Xiong et al., 2022), Box^{2}EL (Jackermeier et al., 2024), and TransBox (Yang et al., 2025) embed $\mathcal{EL}_\bot$ knowledge bases with provable soundness (Chen et al., 2025) but produce crisp membership; FUZZYCELL uses fuzzy membership. DeepGO-SE (Kulmanov et al., 2024) is the closest prior work, using $\mathcal{EL}_\bot$ axioms from the Gene Ontology for protein function prediction; FUZZYCELL transfers this idea to cell type annotation (CL) with Sigmoidal Reichenbach scoring. The retrieve-then-rerank paradigm from biomedical entity linking (Jin et al., 2023) inspires FUZZYCELL's two-stage design.

6 CONCLUSION AND FUTURE WORK

We introduced FUZZYCELL, the first framework to integrate differentiable fuzzy $\mathcal{EL}_\bot$ reasoning into single-cell annotation. By resolving the implication bias via the Sigmoidal Reichenbach implication, FUZZYCELL produces graded membership vectors that are biologically meaningful and provably consistent with the Cell Ontology. Across three benchmarks, FUZZYCELL reduces axiom violations by over 80% while improving Macro-F1 by 4–8 points over the respective backbone, with the largest gains on rare and zero-shot types, as a backbone-agnostic plug-in adding $<$4 ms per cell.

Immediate Future Work. Beyond single-cell annotation, the retrieve-then-reason paradigm could be applied to any OWL-backed annotation task, including gene and protein function prediction (Gene Ontology, Protein Ontology, Trait Ontology, Plant Ontology), disease phenotyping (Human Phenotype Ontology, Human Disease Ontologies, Orphanet Rare Disease Ontology), and spatial transcriptomic cell typing. Integrating pathway-level knowledge (e.g., Reactome) as additional axioms could further improve disambiguation of functionally similar cell types. A detailed discussion of additional limitations and broader impact is provided in Section F.

REPRODUCIBILITY STATEMENT

Code for FUZZYCELL, including the retrieval module, the differentiable $\mathcal{EL}_\perp$ reasoner, the axiom scorer, training and evaluation scripts, and configuration files for all reported experiments, is available at `https://anonymous.4open.science/r/FuzzyCell`. The TBox extraction pipeline and the exact ontology releases used (Cell Ontology 2024-04-05 and Uberon 2024-03-22) are documented in Section D.1. Backbone configurations and checkpoints are specified in Section D.2; the module architecture, parameter counts, and training procedure in Section D.3 and Section D.4; and the full hyperparameter search space and selected values in Section D.5. Baseline implementations, including the re-implementation of HierCE for which no public code is available, are described in Section D.6. The definitions of all evaluation metrics, including AVR and HD-F1, are given in Section C. Compute resources and total GPU hours are reported in Section D.7. Complete proofs of all theoretical claims, with all assumptions stated explicitly, are provided in Section A. All results are reported as mean and standard deviation over five independent runs with fixed seeds.

AI USE STATEMENT

During the preparation of this paper, the authors used generative AI tools for copyediting, grammar checking, figure making, and improvements to clarity and readability. They were not used to develop the theoretical framework, to prove any of the theorems or propositions, to propose or refine hypotheses, to design the experiments or to provide feedback on the methodology, to implement the method, to generate or clean any dataset, or to interpret results. All content was developed and finalized by the authors, who take full responsibility for the accuracy and integrity of this paper.

ETHICS STATEMENT

This work uses only publicly available, de-identified single-cell transcriptomic atlases (Tabula Sapiens, HLCA, and AIDA v2) and publicly released ontologies from the OBO Foundry; no new human subjects data were collected, and no IRB approval was required. The method operates on gene expression profiles and ontology axioms and does not generate synthetic biological data. The primary societal risk is over-reliance on automated annotation in downstream biological or clinical analyses: because the graded membership scores have so far been validated only against model-derived pseudotime rather than against expert-curated or experimentally measured transitional states, they should not be used as a substitute for expert review in settings where annotation errors have consequences. We also note that the ontology axioms used as constraints reflect the current state of the Cell Ontology, which may under-represent cell types from populations that are less studied, and that annotation systems constrained by such axioms could inherit that coverage bias. A fuller discussion of limitations and broader impact appears in Section F.

REFERENCES

Regev A, Teichmann SA, Lander ES, Amit I, Benoist C, Birney E, Bodenmiller B, Campbell P, Carninci P, Clatworthy M, Clevers H, Deplancke B, Dunham I, Eberwine J, Eils R, Enard W, Farmer A, Fugger L, Göttgens B, Hacohen N, Haniffa M, Hemberg M, Kim S, Klenerman P, Kriegstein A, Lein E, Linnarsson S, Lundberg E, Lundeberg J, Majumder P, Marioni JC, Merad M, Mhlanga M, Nawijn M, Netea M, Nolan G, Pe'er D, Phillipakis A, Ponting CP, Quake S, Reik W, Rozenblatt-Rosen O, Sanes J, Satija R, Schumacher TN, Shalek A, Shapiro E, Sharma P, Shin JW, Stegle O, Stratton M, Stubbington MJT, Theis FJ, van Oudenaarden A Uhlen M, Wagner A, Watt F, Weissman J, Wold B, Xavier R, and Yosef N; Human Cell Atlas Meeting Participants. Science Forum: The Human Cell Atlas. *eLife*, 6:e27041, 2017.

Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M. Bell, Emanuele Bezzi, et al. CZ CELLxGENE Discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic Acids Res.*, 53(D1):D886–D900, 2025.

Dvir Aran, Agnieszka P. Looney, Leqian Liu, Esther Wu, Valerie Fong, Austin Hsu, et al. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nat. Immunol.*, 20(2):163–172, 2019.

Franz Baader, Sebastian Brandt, and Carsten Lutz. Pushing the $\mathcal{EL}$ Envelope. In *Proc. IJCAI'05*, pp. 364–369. Professional Book Center, 2005.

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

Samy Badreddine, Artur S. d'Avila Garcez, Luciano Serafini, and Michael Spranger. Logic Tensor Networks. *Artif. Intell.*, 303:103649, 2022.

Jonathan Bard, Seung Y. Rhee, and Michael Ashburner. An ontology for cell types. *Genome Biol.*, 6 (2):R21, 2005.

Sean Bechhofer. OWL: Web Ontology Language. In *Encyclopedia of Database Systems, Second Edition*. Springer, 2018.

Volker Bergen, Marius Lange, Stefan Peidli, F. Alexander Wolf, and Fabian J. Theis. Generalizing RNA velocity to transient cell states through dynamical modeling. *Nat. Biotechnol.*, 38(12): 1408–1414, 2020.

Matthew N. Bernstein, Zhongjie Ma, Michael Gleicher, and Colin N. Dewey. CellO: Comprehensive and Hierarchical Cell Type Classification of Human Cells with the Cell Ontology. *iScience*, 24(1): 101913, 2021.

Rebecca Boiarsky, Nalini M. Singh, Alejandro Buendia, Ava P. Amini, Gad Getz, and David Sontag. Deeper evaluation of a single-cell foundation model. *Nat. Mach. Intell.*, 6:1443–1446, 2024.

Jiaoyan Chen, Olga Mashkova, Fernando Zhapa-Camacho, Robert Hoehndorf, Yuan He, and Ian Horrocks. Ontology Embedding: A Survey of Methods, Applications and Resources. *IEEE Trans. Knowl. Data Eng.*, 37(7):4193–4212, 2025.

Seewon Choi, Alaia Solko-Breslin, Rajeev Alur, and Eric Wong. CTSketch: Compositional Tensor Sketching for Scalable Neurosymbolic Learning. In *Advances in Neural Information Processing Systems 38 (NeurIPS'25)*, 2025.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scGPT: toward building a foundation model for single-cell multi-omics using generative AI. *Nat. Methods*, 21(8):1470–1480, 2024.

Sebastiano Cultrera di Montesano, Davide D'Ascenzo, Srivatsan Raghavan, Ava P. Amini, Peter S. Winter, and Lorin Crawford. Improving atlas-scale single-cell annotation models with hierarchical cross-entropy loss. *Nat. Comput. Sci.*, 6:243–249, 2026.

Alexander D. Diehl, Terrence F. Meehan, Yvonne M. Bradford, Matthew H. Brush, Wasila M. Dahdul, David S. Dougall, Yongqun He, David Osumi-Sutherland, Alan Ruttenberg, Sirarat Sarntivijai, Ceri E. Van Slyke, Nicole A. Vasilevsky, Melissa A. Haendel, Judith A. Blake, and Christopher J. Mungall. The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability. *J. Biomed. Semant.*, 7(1):44, 2016.

Michelangelo Diligenti, Marco Gori, and Claudio Saccà. Semantic-based regularization for learning and inference. *Artif. Intell.*, 244:143–165, 2017.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing. *ACM Trans. Comput. Heal.*, 3(1):2:1–2:23, 2022.

Hao-Yuan He, Wang-Zhou Dai, and Ming Li. Reduced implication-bias logic loss for neuro-symbolic learning. *Mach. Learn.*, 113(6):3357–3377, 2024.

Robert Hoehndorf, Michel Dumontier, Anika Oellrich, Sarala Wimalaratne, Dietrich Rebholz-Schuhmann, Paul Schofield, and Georgios V. Gkoutos. A common layer of interoperability for biomedical ontologies based on OWL EL. *Bioinformatics*, 27(7):1001–1008, 2011.

Mathias Jackermeier, Jiaoyan Chen, and Ian Horrocks. Dual Box Embeddings for the Description Logic $\mathcal{EL}^{++}$. In *Proc. WWW'24*, pp. 2250–2258. ACM, 2024.

Qiao Jin, Won Kim, Qingyu Chen, Donald C. Comeau, Lana Yeganova, W. John Wilbur, and Zhiyong Lu. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinform.*, 39(10), 2023.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-Scale Similarity Search with GPUs. *IEEE Trans. Big Data*, 7(3):535–547, 2021.

Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancík. The Incredible ELK - From Polynomial Procedures to Efficient Reasoning with $\mathcal{EL}$ Ontologies. *J. Autom. Reason.*, 53(1):1–61, 2014.

Kasia Z. Kedzierska, Lorin Crawford, Ava P. Amini, and Alex X. Lu. Zero-shot evaluation reveals limitations of single-cell foundation models. *Genome Biol.*, 26(1):101, 2025.

Maxat Kulmanov, Wang Liu-Wei, Yuan Yan, and Robert Hoehndorf. EL Embeddings: Geometric Construction of Models for the Description Logic $\mathcal{EL}^{++}$. In *Proc. IJCAI'19*, pp. 6103–6109. ijcai.org, 2019.

Maxat Kulmanov, Francisco J. Guzmán-Vega, Paula Duek Roggli, Lydie Lane, Stefan T. Arold, and Robert Hoehndorf. Protein function prediction as approximate semantic entailment. *Nat. Mac. Intell.*, 6(2):220–228, 2024.

Constance Le Gac, Alice Driessen, Nicolas Deutschmann, and María Rodríguez Martínez. Fuzzy Logic for Biological Networks as ML Regression: Scaling to Single-Cell Datasets With Autograd. In *NeurIPS 2022 Workshop on Learning Meaningful Representations of Life (LMRL)*, 2022.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. Self-Alignment Pretraining for Biomedical Entity Representations. In *Proc. NAACL-HLT'21*, pp. 4228–4238. Association for Computational Linguistics, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *Proc. ICLR'19*. OpenReview.net, 2019.

Jaron Maene, Vincent Derkinderen, and Luc De Raedt. On the Hardness of Probabilistic Neurosymbolic Learning. In *Proc. ICML'24*, Proceedings of Machine Learning Research, pp. 34203–34218. PMLR / OpenReview.net, 2024.

Jaron Maene, Vincent Derkinderen, and Pedro Zuidberg Dos Martires. KLay: Accelerating Arithmetic Circuits for Neurosymbolic AI. In *Proc. ICLR'25*. OpenReview.net, 2025.

Nimish Magre, Ebtisam Alshehri, Fedor Grab, Yerdos Ordabayev, Steven A. McCarroll, Mehrtash Babadi, and Stephen J. Fleming. A Supervised Ontology-Aware Cell Annotation Method for Single-Cell Transcriptomic Data. *bioRxiv*, 2026.

Gioele La Manno, Ruslan Soldatov, Amit Zeisel, Emelie Braun, Hannah Hochgerner, Viktor Petukhov, Katja Lidschreiber, Maria E. Kastriti, Peter Lönnerberg, Alessandro Furlan, others, and Peter V. Kharchenko. RNA velocity of single cells. *Nature*, 560(7719):494–498, 2018.

Aaditya Naik, Jason Liu, Claire Wang, Amish Sethi, Saikat Dutta, Mayur Naik, and Eric Wong. DOLPHIN: A Programmable Framework for Scalable Neurosymbolic Learning. In *Proc. ICML'25*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025.

Lisa Sikkema, Ciro Ramírez-Suástegui, Daniel C. Strobl, Tessa E. Gillett, Luke Zappia, Elo Madissoon, Nikolay S. Markov, Laure-Emmanuelle Zaragosi, Yuge Ji, Meshal Ansari, Marie-Jeanne Arguel, Leonie Apperloo, Martin Banchero, Christophe Bécavin, Marijn Berg, Evgeny Chichelnitskiy, Mei i Chung, Antoine Collin, Aurore C. A. Gay, Janine Gote-Schniering, Baharak Hooshiar Kashani, Kemal Inecik, Manu Jain, Theodore S. Kapellos, Tessa M. Kole, Sylvie Leroy, Christoph H. Mayr, Amanda J. Oliver, Michael von Papen, Lance Peter, Chase J. Taylor, Thomas Walzthoeni, Chuan Xu, Linh T. Bui, Carlo De Donno, Leander Dony, Alen Faiz, Minzhe Guo, Austin J. Gutierrez, Lukas Heumos, Ni Huang, Ignacio L. Ibarra, Nathan D. Jackson, Preetish Kadur Lakshminarasimha Murthy, Mohammad Lotfollahi, Tracy Tabib, Carlos Talavera-López, Kyle J. Travaglini, Anna Wilbrey-Clark, Kaylee B. Worlock, Masahiro Yoshida, Lung Biological Network Consortium, Maarten van den Berge, Yohan Bossé, Tushar J. Desai, Oliver Eickelberg, Naftali Kaminski, Mark A. Krasnow, Robert Lafyatis, Marko Z. Nikolic, Joseph E. Powell, Jayaraj

Rajagopal, Mauricio Rojas, Orit Rozenblatt-Rosen, Max A. Seibold, Dean Sheppard, Douglas P. Shepherd, Don D. Sin, Wim Timens, Alexander M. Tsankov, Jeffrey Whitsett, Yan Xu, Nicholas E. Banovich, Pascal Barbry, Thu Elizabeth Duong, Christine S. Falk, Kerstin B. Meyer, Jonathan A. Kropski, Dana Pe'er, Herbert B. Schiller, Purushothama Rao Tata, Joachim L. Schultze, Sara A. Teichmann, Alexander V. Misharin, Martijn C. Nawijn, Malte D. Luecken, and Fabian J. Theis. An integrated cell atlas of the lung in health and disease. *Nat. Med.*, 29:1563–1577, 2023.

Barry Smith, Michael Ashburner, Cornelius Rosse, Jonathan Bard, William Bug, Werner Ceusters, Louis J Goldberg, Karen Eilbeck, Amelia Ireland, Christopher J Mungall, The OBI Consortium, Neocles Leontis, Philippe Rocca-Serra, Alan Ruttenberg, Susanna-Assunta Sansone, Richard H Scheuermann, Nigam Shah, Patricia L Whetzel, and Suzanna Lewis. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nat. Biotechnol.*, 25(11): 1251–1255, 2007.

Shawn Zheng Kai Tan, Aleix Puig-Barbe, Damien Goutte-Gattat, Caroline Eastwood, Brian Aevermann, Alida Avola, James P. Balhoff, Ismail Ugur Bayindir, Jasmine Belfiore, Anita Reane Caron, David S. Fischer, Nancy George, Benjamin M. Gyori, Melissa A. Haendel, Charles Tapley Hoyt, Huseyin Kir, Tiago Lubiana, Nicolas Matentzoglu, James A. Overton, Beverly Peng, Bjoern Peters, Ellen M. Quardokus, Patrick L. Ray, Paola Roncaglia, Andrea D. Rivera, Ray Stefancsik, Wei Kheng Teh, Sabrina Toro, Nicole Vasilevsky, Chuan Xu, Yun Zhang, Richard H. Scheuermann, Christopher J. Mungall, Alexander D. Diehl, and David Osumi-Sutherland. The Cell Ontology in the age of single-cell omics. *J. Biomed. Semant.*, 2026.

Zhenwei Tang, Tilman Hinnerichs, Xi Peng, Xiangliang Zhang, and Robert Hoehndorf. F$\mathcal{ALC}$ON: Sound and Complete Neural Semantic Entailment over $\mathcal{ALC}$ Ontologies. *CoRR*, abs/2208.07628, 2022.

The Tabula Sapiens Consortium. The Tabula Sapiens: A multiple-organ, single-cell transcriptomic atlas of humans. *Science*, 376(6594):eabl4896, 2022.

Christina V. Theodoris, Ling Xiao, Anant Chopra, Mark D. Chaffin, Zeina R. Al Sayed, Matthew C. Hill, Helene Mantineo, Elizabeth M. Brydon, Zexian Zeng, X. Shirley Liu, and Patrick T. Ellinor. Transfer learning enables predictions in network biology. *Nature*, 618(7965):616–624, 2023.

Cole Trapnell, Davide Cacchiarelli, Jonna Grimsby, Prapti Pokharel, Shuqiang Li, Michael Morse, Niall J Lennon, Kenneth J Livak, Tarjei S Mikkelsen, and John L Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.*, 32(4):381–386, 2014.

Emile van Krieken, Erman Acar, and Frank van Harmelen. Analyzing Differentiable Fuzzy Implications. In *Proc. KR'20*, pp. 893–903, 2020.

Emile van Krieken, Erman Acar, and Frank van Harmelen. Analyzing differentiable fuzzy logic operators. *Artif. Intell.*, 302:103602, 2022.

Sheng Wang, Angela Oliveira Pisco, Aaron McGeever, Maria Brbic, Marinka Zitnik, Spyros Darmanis, Jure Leskovec, Jim Karkanias, and Russ B. Altman. Leveraging the Cell Ontology to classify unseen cell types. *Nat. Commun.*, 12(1):5556, 2021.

F. Alexander Wolf, Philipp Angerer, and Fabian J. Theis. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.*, 19(1):15, 2018.

F. Alexander Wolf, Fiona K. Hamey, Mireya Plass, Jordi Solana, Joakim S. Dahlin, Berthold Göttgens, Nikolaus Rajewsky, Lukas Simon, and Fabian J. Theis. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.*, 20(1):59, 2019.

Xuan Wu and Yizheng Zhao. A Neuro-Symbolic Approach to Symbol Grounding for $\mathcal{ALC}$-Ontologies. In *Proc. KDD'25*, pp. 3240–3249. ACM, 2025.

Bo Xiong, Nico Potyka, Trung-Kien Tran, Mojtaba Nayyeri, and Steffen Staab. Faithful Embeddings for $\mathcal{EL}^{++}$ Knowledge Bases. In *Proc. ISWC'22*, volume 13489 of *LNCS*, pp. 22–38. Springer, 2022.

Chenling Xu, Romain Lopez, Edouard Mehlman, Jeffrey Regier, Michael I. Jordan, and Nir Yosef. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. *Mol. Syst. Biol.*, 17(1):e9620, 2021.

Jingyi Xu, Zilu Zhang, Tal Friedman, Yitao Liang, and Guy Van den Broeck. A Semantic Loss Function for Deep Learning with Symbolic Knowledge. In *Proc. ICML'18*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5498–5507. PMLR, 2018.

Hui Yang, Jiaoyan Chen, and Uli Sattler. TransBox: $\mathcal{EL}^{++}$-closed Ontology Embedding. In *Proc. WWW'25*, pp. 22–34. ACM, 2025.

Xinyu Yuan, Zhihao Zhan, Zuobai Zhang, Manqi Zhou, Jianan Zhao, Boyu Han, Yue Li, and Jian Tang. Cell ontology guided transcriptome foundation model. In *Advances in Neural Information Processing Systems 38 (NeurIPS'24)*, 2024.

Yuansong Zeng, Jiancong Xie, Ningyuan Shangguan, Zhuoyi Wei, Wenbing Li, Yun Su, and Yuedong Yang. CellFM: a large-scale foundation model pre-trained on transcriptomics of 100 million human cells. *Nat. Commun.*, 16:4679, 2025.

Allen W. Zhang, Ciara O'Flanagan, Elizabeth A. Chavez, Jamie L. P. Lim, Nicholas Ceglia, Andrew McPherson, et al. Probabilistic cell-type assignment of single-cell RNA-seq for tumor microenvironment profiling. *Nat. Methods*, 16(10):1007–1015, 2019.

Supplementary Material: Table of Contents

Appendix A Full Proofs 16

A.1 Proof of Proposition 3.1 (Gradient Non-Degeneracy) 16

A.2 Proof of Theorem 3.2 (Semantic Soundness) 17

A.3 Additional Theoretical Results 18

Appendix B Additional Experimental Results 20

B.1 Results on HLCA and AIDA v2 20

B.2 Per-Tissue Breakdown on Tabula Sapiens 21

B.3 Zero-Shot Cell Type Performance 21

B.4 Qualitative Case Studies (8 Cases) 22

B.5 Extended Ablation Experiments 24

B.6 Fuzzy vs. Crisp Ontological Reasoning 27

B.7 Scalability Analysis 29

Appendix C Ontology-Aware Evaluation Metrics 32

C.1 Axiom Violation Rate (AVR) 32

C.2 Hierarchy-Distance-Weighted F1 (HD-F1) 34

Appendix D Implementation Details 35

D.1 TBox Construction and Preprocessing 35

D.2 Backbone Configuration 36

D.3 FuzzyCell Module Architecture 37

D.4 Training Details 37

D.5 Hyperparameter Search 38

D.6 Baseline Implementation Details 38

D.7 Compute Resources and Carbon Footprint 39

Appendix E Extended Related Work 39

Appendix F Limitations and Broader Impact 40

A FULL PROOFS

We provide complete proofs for all theoretical results stated in Section 3.4. Throughout, we use the following notation: $\sigma(z) = (1 + e^{-z})^{-1}$ denotes the logistic sigmoid, $\sigma'(z) = \sigma(z)(1 - \sigma(z))$ its derivative, $I_R(a, b) = 1 - a + ab$ the Reichenbach implication, and $I_\sigma(a, b) = \sigma\big(s \cdot (I_R(a, b) - \frac{1}{2})\big)$ the Sigmoidal Reichenbach implication with steepness parameter $s > 0$. We write the logit as $z(a, b) = s \cdot (I_R(a, b) - \frac{1}{2}) = s(\frac{1}{2} - a + ab)$.

A.1 PROOF OF PROPOSITION 3.1 (GRADIENT NON-DEGENERACY)

We restate the proposition for convenience.

Proposition (Gradient non-degeneracy)**.** Let $I_\sigma(a, b) = \sigma\big(s \cdot (I_R(a, b) - \frac{1}{2})\big)$ with $s > 0$. Then:

(i) $\|\nabla I_\sigma(a, b)\| > 0$ for all $(a, b) \in (0, 1)^2$.
(ii) In contrast, the Goguen implication $I_{GG}(a, b) = \min(1, b/a)$ satisfies $\nabla I_{GG} = \mathbf{0}$ for all $a \le b$, covering at least 50% of $[0, 1]^2$. The standard Reichenbach I_R has $\partial I_R/\partial a = b - 1$, which vanishes on the entire line $b = 1$.

Proof. We organize the proof into four parts.

Part 1: Partial derivatives of I_σ. Since $I_\sigma(a, b) = \sigma(z)$ with $z = s(\frac{1}{2} - a + ab)$, the chain rule gives:

$$\frac{\partial I_\sigma}{\partial a} = \sigma'(z) \cdot \frac{\partial z}{\partial a} = \sigma'(z) \cdot s \cdot \frac{\partial}{\partial a}\big(\tfrac{1}{2} - a + ab\big) = s\,\sigma'(z)\,(b - 1), \tag{3}$$

$$\frac{\partial I_\sigma}{\partial b} = \sigma'(z) \cdot \frac{\partial z}{\partial b} = \sigma'(z) \cdot s \cdot \frac{\partial}{\partial b}\big(\tfrac{1}{2} - a + ab\big) = s\,\sigma'(z)\,a. \tag{4}$$

We now verify the signs of each factor in these expressions for $(a, b) \in (0, 1)^2$:

- $s > 0$ by assumption.
- $\sigma'(z) = \sigma(z)(1 - \sigma(z))$. Since $\sigma(z) \in (0, 1)$ for all finite $z \in \mathbb{R}$, both $\sigma(z)$ and $1 - \sigma(z)$ are strictly positive, so $\sigma'(z) > 0$.
- In (3): since $b \in (0, 1)$, we have $b - 1 \in (-1, 0)$, so $b - 1 < 0$ strictly. Hence $\frac{\partial I_\sigma}{\partial a} = s \cdot \sigma'(z) \cdot (b - 1) < 0$.
- In (4): since $a \in (0, 1)$, we have $a > 0$ strictly. Hence $\frac{\partial I_\sigma}{\partial b} = s \cdot \sigma'(z) \cdot a > 0$.

Both partial derivatives are strictly nonzero, with $\frac{\partial I_\sigma}{\partial a} < 0$ (increasing the antecedent decreases the implication value) and $\frac{\partial I_\sigma}{\partial b} > 0$ (increasing the consequent increases it), consistent with the expected behavior of a logical implication.

Part 2: Positivity of the gradient norm. The squared gradient norm is:

$$\begin{aligned}\|\nabla I_\sigma(a, b)\|^2 &= \Big(\frac{\partial I_\sigma}{\partial a}\Big)^2 + \Big(\frac{\partial I_\sigma}{\partial b}\Big)^2 \\ &= \big[s\,\sigma'(z)\,(b - 1)\big]^2 + \big[s\,\sigma'(z)\,a\big]^2 \\ &= s^2\big[\sigma'(z)\big]^2\big[(b - 1)^2 + a^2\big].\end{aligned} \tag{5}$$

Each factor is strictly positive on $(0, 1)^2$: $s^2 > 0$ by assumption; $[\sigma'(z)]^2 > 0$ since $\sigma'(z) > 0$ as shown above; $(b - 1)^2 > 0$ since $b \neq 1$, and $a^2 > 0$ since $a \neq 0$, so their sum is strictly positive. Therefore $\|\nabla I_\sigma(a, b)\|^2 > 0$, which implies $\|\nabla I_\sigma(a, b)\| > 0$ for all $(a, b) \in (0, 1)^2$. This completes the proof of (i).

Part 3: Degeneracy of the Goguen implication. The Goguen (residual) implication for the product t-norm is defined as:

$$I_{GG}(a, b) = \begin{cases} 1 & \text{if } a \le b, \\ b/a & \text{if } a > b. \end{cases}$$

Consider the region $R_{\text{cons}} = \{(a, b) \in (0, 1)^2 : a \leq b\}$. On this region, $I_{GG}(a, b) = 1$ is constant, so $\nabla I_{GG}(a, b) = \mathbf{0}$ everywhere in R_{cons}. The area of R_{cons} within the unit square $[0, 1]^2$ is $\int_0^1 \int_0^b 1 \, da \, db = \int_0^1 b \, db = \frac{1}{2}$, confirming that the zero-gradient region covers exactly 50% of $[0, 1]^2$.

This degeneracy is problematic for learning because the region $a \leq b$ corresponds precisely to the logically *consistent* pairs, where the antecedent membership does not exceed the consequent membership. A model whose current parameters produce logically consistent outputs receives no gradient signal from Goguen-based axiom losses, preventing further refinement of concept boundaries.

Part 4: Degeneracy of the standard Reichenbach implication. The Reichenbach implication $I_R(a, b) = 1 - a + ab$ has partial derivatives:

$$\frac{\partial I_R}{\partial a} = b - 1, \qquad \frac{\partial I_R}{\partial b} = a.$$

The gradient $\frac{\partial I_R}{\partial a}$ vanishes on the entire line $b = 1$, and $\frac{\partial I_R}{\partial b}$ vanishes on $a = 0$. While the zero-gradient *set* has measure zero (a line rather than a region), the gradient magnitude $|b-1|$ is continuous, so for b close to 1 the gradient becomes arbitrarily small. This is the *implication bias* identified by van Krieken et al. (2020): when the consequent has high membership ($b \approx 1$), the gradient with respect to the antecedent nearly vanishes, incentivizing the optimizer to inflate all membership degrees rather than learn meaningful type boundaries.

The Sigmoidal Reichenbach I_σ resolves this by applying a nonlinear amplification via σ. Although the factor $(b - 1)$ still appears in (3), it is multiplied by $\sigma'(z)$, which can be large when z is near zero (i.e., near the decision boundary $I_R = \frac{1}{2}$). The net effect is that I_σ redistributes gradient mass toward the decision boundary, where disambiguation matters most, rather than allowing it to vanish at $b \approx 1$. □

Remark: Discrimination growth. Although not part of the formal proposition, we note a useful quantitative consequence. Consider a hard negative with $(a, b) = (0.9, 0.1)$ and a hard positive with $(a, b) = (0.5, 0.9)$. Their logits are:

$$z_{\text{neg}} = s(\tfrac{1}{2} - 0.9 + 0.9 \times 0.1) = s(\tfrac{1}{2} - 0.9 + 0.09) = -0.31s,$$
$$z_{\text{pos}} = s(\tfrac{1}{2} - 0.5 + 0.5 \times 0.9) = s(\tfrac{1}{2} - 0.5 + 0.45) = 0.45s.$$

The discrimination ratio is:

$$\text{DR}(s) = \frac{I_\sigma(0.5, 0.9)}{I_\sigma(0.9, 0.1)} = \frac{\sigma(0.45s)}{\sigma(-0.31s)} = \frac{1 + e^{0.31s}}{1 + e^{-0.45s}}.$$

For large s, the denominator approaches 1 and the numerator is dominated by $e^{0.31s}$, so $\text{DR}(s) \sim e^{0.31s}$. Substituting specific values: $\text{DR}(10) \approx e^{3.1} \approx 22$, and $\text{DR}(20) \approx e^{6.2} \approx 493$. By contrast, the linear Reichenbach implication gives a constant ratio $I_R(0.5, 0.9)/I_R(0.9, 0.1) = 0.95/0.19 = 5$, independent of any parameter. This exponential growth in discrimination provides the mechanism by which ontological signals can override strong neural similarity for type-inconsistent candidates.

A.2 Proof of Theorem 3.2 (Semantic Soundness)

We restate the theorem for convenience.

Theorem (Semantic soundness)**.** The fuzzy semantics induced by I_σ is a conservative extension of classical $\mathcal{EL}_\bot$:

(i) **Boundary preservation.** On crisp inputs $a, b \in \{0, 1\}$: $\lim_{s \to \infty} I_\sigma(a, b) = \mathbb{1}[a \leq b]$.
(ii) **Soundness.** If $\mathcal{T}_{\mathsf{CL}} \models C \sqsubseteq D$, then for any crisp interpretation $\mathcal{I}$: $\lim_{s \to \infty} \inf_{d \in \Delta^{\mathcal{I}}} I_\sigma\big(C^{\mathcal{I}}(d),\ D^{\mathcal{I}}(d)\big) = 1$.

Proof. We proceed in three steps.

Step 1: Asymptotic properties of the sigmoid. For any constant $\delta > 0$:

$$\lim_{s\to\infty} \sigma(s \cdot \delta) = \lim_{s\to\infty} \frac{1}{1+e^{-s\delta}} = \frac{1}{1+0} = 1, \tag{6}$$

$$\lim_{s\to\infty} \sigma(-s \cdot \delta) = \lim_{s\to\infty} \frac{1}{1+e^{s\delta}} = \frac{1}{1+\infty} = 0. \tag{7}$$

These follow directly from the definition of the logistic sigmoid and the behavior of the exponential function.

Step 2: Boundary preservation (proof of (i)). We analyze all four cases for $(a, b) \in \{0, 1\}^2$. For each case, we compute $I_R(a, b)$, the logit $z = s(I_R - \frac{1}{2})$, and the limit of $I_\sigma = \sigma(z)$ as $s \to \infty$.

a	b	$I_R(a,b) = 1 - a + ab$	$I_R - \frac{1}{2}$	$\lim_{s\to\infty} I_\sigma$	Classical $a \to b$
0	0	$1 - 0 + 0 = 1$	$+0.5$	$\sigma(0.5s) \to 1$	1
0	1	$1 - 0 + 0 = 1$	$+0.5$	$\sigma(0.5s) \to 1$	1
1	1	$1 - 1 + 1 = 1$	$+0.5$	$\sigma(0.5s) \to 1$	1
1	0	$1 - 1 + 0 = 0$	-0.5	$\sigma(-0.5s) \to 0$	0

In all four cases, $\lim_{s\to\infty} I_\sigma(a, b)$ matches the truth table of classical Boolean implication $\not\Vdash[a \leq b]$:

- When $(a, b) \in \{(0, 0), (0, 1), (1, 1)\}$: $I_R = 1 > \frac{1}{2}$, so the logit is $0.5s \to +\infty$, and $\sigma \to 1$ by (6).
- When $(a, b) = (1, 0)$: $I_R = 0 < \frac{1}{2}$, so the logit is $-0.5s \to -\infty$, and $\sigma \to 0$ by (7).

This establishes boundary preservation.

Step 3: Soundness (proof of (ii)). Assume $\mathcal{T}_{\mathsf{CL}} \models C \sqsubseteq D$ in the classical sense. Let $\mathcal{I}$ be any crisp interpretation (i.e., one where $A^{\mathcal{I}}(d) \in \{0, 1\}$ for all concept names A and all $d \in \Delta^{\mathcal{I}}$) that is a model of $\mathcal{T}_{\mathsf{CL}}$. By the definition of classical entailment, $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$, which means that for every $d \in \Delta^{\mathcal{I}}$:

$$C^{\mathcal{I}}(d) = 1 \implies D^{\mathcal{I}}(d) = 1.$$

Equivalently, the pair $(C^{\mathcal{I}}(d), D^{\mathcal{I}}(d))$ can only take the values $(0, 0)$, $(0, 1)$, or $(1, 1)$; the case $(1, 0)$ is excluded by the entailment.

From the pointwise analysis in Step 2, for each of the three admissible cases:

$$I_R(C^{\mathcal{I}}(d), D^{\mathcal{I}}(d)) = 1 > \tfrac{1}{2},$$

so the logit $z(d) = s(I_R - \frac{1}{2}) = 0.5s > 0$ for all $d \in \Delta^{\mathcal{I}}$.

Therefore:

$$\lim_{s\to\infty} I_\sigma\big(C^{\mathcal{I}}(d),\ D^{\mathcal{I}}(d)\big) = \lim_{s\to\infty} \sigma(0.5s) = 1 \qquad \text{for every } d \in \Delta^{\mathcal{I}}.$$

Since the limit equals 1 for every domain element, the infimum also converges to 1:

$$\lim_{s\to\infty} \inf_{d\in\Delta^{\mathcal{I}}} I_\sigma\big(C^{\mathcal{I}}(d),\ D^{\mathcal{I}}(d)\big) = 1.$$

This completes the proof. □

A.3 Additional Theoretical Results

We establish three supplementary properties of I_σ that, while not required for the main results, provide further insight into its behavior and are useful for implementation.

Lemma A.1 (Monotonicity). *For all $s > 0$ and $(a, b) \in (0, 1)^2$:*

(i) $I_\sigma(a, b)$ is strictly decreasing in a (the antecedent).
(ii) $I_\sigma(a, b)$ is strictly increasing in b (the consequent).
(iii) $I_\sigma(a, b)$ is strictly increasing in s whenever $I_R(a, b) > \frac{1}{2}$, and strictly decreasing in s whenever $I_R(a, b) < \frac{1}{2}$.

Proof. Parts (i) and (ii) follow directly from the signs of the partial derivatives established in (3) and (4): $\frac{\partial I_\sigma}{\partial a} < 0$ on $(0,1)^2$ and $\frac{\partial I_\sigma}{\partial b} > 0$ on $(0,1)^2$.

For part (iii), we compute:

$$\frac{\partial I_\sigma}{\partial s} = \sigma'(z) \cdot \frac{\partial z}{\partial s} = \sigma'(z) \cdot (I_R(a,b) - \tfrac{1}{2}).$$

Since $\sigma'(z) > 0$, the sign of $\frac{\partial I_\sigma}{\partial s}$ is determined by the sign of $I_R(a,b) - \frac{1}{2}$. When $I_R > \frac{1}{2}$ (logically consistent pair), increasing s pushes I_σ toward 1. When $I_R < \frac{1}{2}$ (logically inconsistent pair), increasing s pushes I_σ toward 0. This confirms the intuition that larger s sharpens the logical discrimination. □

Lemma A.2 (Value range and centering)**.** *$I_\sigma(a,b) = \frac{1}{2}$ if and only if $I_R(a,b) = \frac{1}{2}$, i.e., $a(1-b) = \frac{1}{2}$, which defines a curve in $[0,1]^2$ passing through $(\frac{1}{2}, 0)$ and $(1, \frac{1}{2})$. In particular, $I_\sigma(\frac{1}{2}, \frac{1}{2}) = \sigma(s/4) > \frac{1}{2}$ for all $s > 0$.*

Proof. $I_\sigma(a,b) = \sigma(s \cdot (I_R(a,b) - \frac{1}{2})) = \frac{1}{2}$ if and only if $s \cdot (I_R - \frac{1}{2}) = 0$, which, since $s > 0$, holds if and only if $I_R(a,b) = \frac{1}{2}$. Substituting $I_R(a,b) = 1 - a + ab$, the equation $1 - a + ab = \frac{1}{2}$ simplifies to $a(1-b) = \frac{1}{2}$. This curve passes through $(\frac{1}{2}, 0)$ and $(1, \frac{1}{2})$, and is strictly below the diagonal $a = b$ in $(0,1)^2$. For the special case $(a,b) = (\frac{1}{2}, \frac{1}{2})$: $I_R = 1 - \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$, so $I_\sigma(\frac{1}{2}, \frac{1}{2}) = \sigma(s \cdot \frac{1}{4}) > \frac{1}{2}$ since $s/4 > 0$. □

Lemma A.3 (Comparison with Łukasiewicz implication)**.** *The Łukasiewicz implication $I_{ŁK}(a,b) = \min(1, 1 - a + b)$ has $\frac{\partial I_{ŁK}}{\partial a} = 0$ whenever $1 - a + b \geq 1$, i.e., whenever $a \leq b$. This covers the same 50% of $[0,1]^2$ as the Goguen implication, confirming that the gradient degeneracy is not specific to Goguen but is shared by all residual implications of continuous t-norms.*

Proof. $I_{\text{ŁK}}(a,b) = \min(1, 1 - a + b)$. When $a \leq b$, $1 - a + b \geq 1$, so $I_{\text{ŁK}} = 1$ (constant), and $\nabla I_{\text{ŁK}} = \mathbf{0}$. The region $\{(a,b) : a \leq b\}$ has area $\frac{1}{2}$ in $[0,1]^2$, matching the Goguen case.

For completeness, we note that the Gödel implication $I_G(a,b) = 1$ if $a \leq b$ and b otherwise also has zero gradient on $\{a \leq b\}$. Thus, all three standard R-implications (Goguen, Łukasiewicz, Gödel) exhibit gradient degeneracy on at least 50% of the domain, while I_σ maintains nonzero gradients everywhere on $(0,1)^2$. □

Proposition A.4 (Inference complexity)**.** FUZZYCELL*'s inference complexity separates into an offline and an online phase:*

- *(i)* ***Offline:*** *computing candidate type memberships $\tau^{\mathcal{I}_f}(C_j) = \mathbb{1}[\mathcal{T}_{\mathsf{CL}} \models C_j \sqsubseteq \tau]$ for all candidates $C_j \in \mathsf{N_C}$ and semantic types $\tau \in \Gamma$ via classical $\mathcal{EL}_\bot$ reasoning in* PTIME *with respect to $|\mathcal{T}_{\mathsf{CL}}|$.*
- *(ii)* ***Online:*** *re-ranking k candidates requires one encoder forward pass ($\mathcal{O}(d^2)$ for a Transformer with hidden dimension d), one dual projection ($\mathcal{O}(d \cdot d_L)$), type inference ($\mathcal{O}(|\Gamma| \cdot d_L)$), and consistency scoring ($\mathcal{O}(k \cdot |\Gamma|)$).*

With typical values k=64, $|\Gamma|$=128, d=512, d_L=256, the reasoning module adds <4 ms per cell.

Proof. **Offline phase.** Subsumption checking in $\mathcal{EL}_\bot$ is known to be PTIME-complete in $|\mathcal{T}|$ (Baader et al., 2005). The ELK reasoner (Kazakov et al., 2014) precomputes the full subsumption hierarchy of $\mathcal{T}_{\mathsf{CL}}$ in polynomial time. The membership matrix $\mathbf{M} \in \{0,1\}^{|\mathsf{N_C}| \times |\Gamma|}$ is computed once and stored; subsequent lookups for any candidate C_j cost $\mathcal{O}(1)$.

Online phase. Given a cell embedding $\mathbf{z}_{\text{cell}} \in \mathbb{R}^d$ (produced by the TFM backbone at cost $\mathcal{O}(d^2)$), FUZZYCELL's reasoning module performs:

1. *Dual projection:* $\mathbf{z}'_{\text{cell}} = \pi_{\text{cell}}(\mathbf{z}_{\text{cell}}) \in \mathbb{R}^{d_L}$, costing $\mathcal{O}(d \cdot d_L)$. The type projections $\{\mathbf{a}'_\tau\}_{\tau \in \Gamma}$ are precomputed and cached.
2. *Type inference:* computing $\tau^{\mathcal{I}_f}(\mathbf{x}) = \sigma({\mathbf{z}'_{\text{cell}}}^\top \mathbf{a}'_\tau / \theta)$ for all $\tau \in \Gamma$ costs $\mathcal{O}(|\Gamma| \cdot d_L)$.

3. *Consistency scoring:* evaluating $s_{\text{onto}}(\mathbf{x}, C_j) = \prod_\tau I_\sigma(\tau^{\mathcal{I}_f}(\mathbf{x}), \tau^{\mathcal{I}_f}(C_j))$ for k candidates costs $\mathcal{O}(k \cdot |\Gamma|)$, since each I_σ evaluation is $\mathcal{O}(1)$ and $\tau^{\mathcal{I}_f}(C_j)$ is looked up from the precomputed matrix.
4. *Score fusion and argmax:* $\mathcal{O}(k)$.

The total online overhead is $\mathcal{O}(d \cdot d_L + |\Gamma| \cdot d_L + k \cdot |\Gamma|)$. With our default hyperparameters: $512 \times 256 + 128 \times 256 + 64 \times 128 = 131{,}072 + 32{,}768 + 8{,}192 = 172{,}032$ multiply-add operations, which completes in <4 ms on an A100 GPU. This represents an overhead of 8–13% over the TFM backbone encoding, which itself costs 28–45 ms per cell. □

B Additional Experimental Results

B.1 Results on HLCA and AIDA v2

Tables 3 and 4 report the full results on HLCA and AIDA v2, following the same protocol as Table 1.

Table 3: **Results on HLCA** (580K cells, 61 types, 20% zero-shot). Mean ± std over 5 independent runs. All FuzzyCell variants improve over their respective backbones ($p < 0.01$, Wilcoxon signed-rank test with Bonferroni correction).

Category	Method	Acc. ↑	Macro-F1 ↑	W-F1 ↑	AVR (%) ↓	HD-F1 ↑
Classical	SingleR	82.4 ±0.3	54.8 ±0.7	81.0 ±0.3	17.5 ±0.9	64.2 ±0.5
	scANVI	84.1 ±0.3	57.2 ±0.6	82.9 ±0.3	15.3 ±0.8	66.8 ±0.4
	CellAssign	80.8 ±0.4	51.6 ±0.9	79.5 ±0.5	19.7 ±1.1	61.5 ±0.7
TFM (flat)	scGPT	90.1 ±0.2	63.5 ±0.5	89.2 ±0.2	13.8 ±0.7	74.1 ±0.4
	Geneformer	89.3 ±0.2	61.8 ±0.6	88.4 ±0.3	15.2 ±0.8	72.6 ±0.5
	CellFM	88.9 ±0.3	60.4 ±0.7	87.8 ±0.3	16.1 ±0.9	71.5 ±0.5
TFM (onto-aware)	scCello	91.2 ±0.2	65.8 ±0.4	90.3 ±0.2	10.5 ±0.6	76.4 ±0.3
	HierCE	90.7 ±0.2	64.9 ±0.5	89.8 ±0.2	12.3 ±0.7	75.1 ±0.4
	FuzzyCell +scGPT	**92.3** ±0.1	**71.2** ±0.4	**91.6** ±0.2	**1.9** ±0.2	**82.5** ±0.3
	FuzzyCell +Geneformer	91.7 ±0.2	69.5 ±0.4	91.0 ±0.2	2.5 ±0.3	80.8 ±0.3
	FuzzyCell +CellFM	91.3 ±0.2	68.1 ±0.5	90.5 ±0.2	2.8 ±0.3	79.6 ±0.4
	FuzzyCell +scCello	92.1 ±0.1	70.8 ±0.3	91.4 ±0.2	1.7 ±0.2	82.1 ±0.3

Table 4: **Results on AIDA v2** (212K cells, 85 types, 20% zero-shot). Mean ± std over 5 independent runs.

Category	Method	Acc. ↑	Macro-F1 ↑	W-F1 ↑	AVR (%) ↓	HD-F1 ↑
Classical	SingleR	76.5 ±0.4	39.2 ±0.9	74.8 ±0.5	24.3 ±1.3	56.1 ±0.7
	scANVI	78.8 ±0.4	42.5 ±0.8	77.2 ±0.4	21.6 ±1.1	59.0 ±0.6
	CellAssign	74.2 ±0.5	36.1 ±1.2	72.5 ±0.6	27.1 ±1.5	53.2 ±0.9
TFM (flat)	scGPT	85.6 ±0.3	49.4 ±0.7	84.3 ±0.3	20.8 ±0.9	67.2 ±0.5
	Geneformer	84.5 ±0.3	47.1 ±0.8	83.1 ±0.4	22.5 ±1.0	65.4 ±0.6
	CellFM	84.0 ±0.3	46.2 ±0.8	82.6 ±0.4	23.4 ±1.1	64.3 ±0.6
TFM (onto-aware)	scCello	87.1 ±0.2	52.6 ±0.6	85.9 ±0.3	16.4 ±0.8	70.5 ±0.4
	HierCE	86.4 ±0.3	51.2 ±0.7	85.2 ±0.3	18.5 ±0.9	69.0 ±0.5
	FuzzyCell +scGPT	**88.4** ±0.2	**57.8** ±0.5	**87.4** ±0.2	**3.5** ±0.4	**76.2** ±0.4
	FuzzyCell +Geneformer	87.6 ±0.2	55.9 ±0.6	86.5 ±0.3	4.2 ±0.4	74.1 ±0.4
	FuzzyCell +CellFM	87.1 ±0.3	54.7 ±0.6	85.9 ±0.3	4.8 ±0.5	73.0 ±0.5
	FuzzyCell +scCello	88.1 ±0.2	57.2 ±0.5	87.1 ±0.2	3.1 ±0.3	75.8 ±0.3

The pattern is consistent across all three benchmarks. On HLCA, where the 61 cell types are all within the lung, FuzzyCell reduces AVR from 10–16% (baselines) to under 3%, a relative reduction of over 82%. The Macro-F1 gain is particularly pronounced (up to 5 points over scCello), likely because the lung contains several closely related cell types (e.g., alveolar type I vs. type II cells, club cells vs. goblet cells) where ontological constraints help disambiguate. On AIDA v2, which tests cross-dataset generalization, baselines exhibit higher AVR (16–27%) due to distribution shift;

FUZZYCELL consistently brings AVR below 5% while improving Macro-F1 by up to 5 points. The fact that FUZZYCELL's improvement is larger on AIDA v2 (the hardest benchmark) than on Tabula Sapiens confirms that ontological constraints are most valuable when the neural representations are less reliable.

B.2 PER-TISSUE BREAKDOWN ON TABULA SAPIENS

Table 5: **Per-tissue results on Tabula Sapiens** (mean over 5 independent runs). Tissues sorted by baseline AVR (scGPT, descending). FUZZYCELL provides the largest gains on multi-lineage tissues (bone marrow, blood, spleen) where disjointness violations are most common.

	Macro-F1 ↑				AVR (%) ↓			
Tissue	scGPT	scCello	FUZZYCELL +scGPT	FUZZYCELL +scCello	scGPT	scCello	FUZZYCELL +scGPT	FUZZYCELL +scCello
Bone marrow	38.2	42.1	**52.5**	51.8	32.4	22.8	**4.1**	4.5
Blood	41.5	45.3	**54.8**	54.1	28.7	19.5	3.8	**3.2**
Spleen	43.8	47.6	**55.2**	54.9	26.1	18.2	3.5	**3.0**
Thymus	45.2	49.1	**56.3**	55.8	24.3	16.9	3.2	**2.8**
Lymph node	47.6	50.8	**57.4**	56.9	22.8	15.4	**2.9**	3.1
Lung	52.1	55.4	**61.2**	60.7	19.5	13.8	**2.6**	2.8
Liver	54.3	57.2	**62.8**	62.1	18.2	12.5	**2.4**	2.6
Kidney	55.8	58.6	**63.5**	62.9	17.4	11.8	**2.3**	2.5
Pancreas	58.2	60.5	**64.8**	64.3	15.6	10.9	**2.1**	2.3
Small intestine	56.4	59.1	**63.9**	63.4	16.8	11.5	**2.2**	2.4
Large intestine	57.1	59.8	**64.2**	63.7	16.2	11.2	**2.1**	2.3
Heart	60.3	62.4	**66.5**	66.0	14.1	9.8	**1.9**	2.0
Skin	61.5	63.2	**67.1**	66.6	13.5	9.4	1.8	**1.7**
Eye	62.8	64.5	**68.2**	67.8	12.8	8.9	**1.7**	1.8
Prostate	63.4	65.1	**68.8**	68.3	12.1	8.5	1.6	**1.5**
Bladder	64.2	65.8	**69.3**	68.9	11.5	8.1	**1.5**	1.6
Mammary gland	65.1	66.5	**70.0**	69.6	10.8	7.6	**1.4**	1.5
Tongue	66.8	68.2	**71.5**	71.1	9.5	6.8	**1.2**	1.3
Trachea	67.4	68.8	**72.0**	71.6	9.1	6.5	**1.1**	1.2
Vasculature	68.5	69.8	**72.8**	72.4	8.4	6.0	**1.0**	1.1
Muscle	69.2	70.5	**73.4**	73.0	7.8	5.6	**0.9**	1.0
Fat	70.8	71.9	**74.5**	74.1	6.5	4.8	**0.8**	0.9
Uterus	71.2	72.3	**75.0**	74.6	6.1	4.5	**0.7**	0.8
Salivary gland	72.5	73.4	**76.0**	75.7	5.2	3.9	**0.6**	0.7

Several patterns emerge from Table 5. First, baseline AVR varies dramatically across tissues: bone marrow (32.4%), blood (28.7%), and spleen (26.1%) have the highest violation rates, while salivary gland (5.2%) and uterus (6.1%) have the lowest. This disparity reflects the biological composition of each tissue: hematopoietic tissues contain multiple coexisting lineages (myeloid, lymphoid, erythroid) with complex disjointness relationships, while tissues dominated by a single lineage (e.g., salivary gland epithelium) offer fewer opportunities for inter-lineage confusion.

Second, FUZZYCELL's Macro-F1 improvement is strongly correlated with baseline AVR ($r > 0.9$). On bone marrow, the improvement is 14.3 points (38.2 $\rightarrow$ 52.5); on salivary gland, it is 3.5 points (72.5 $\rightarrow$ 76.0). This confirms the hypothesis that ontological constraints are most valuable in tissues where the annotation task is inherently more ambiguous.

Third, the role-restriction axioms in CL, which encode tissue-residency rules (e.g., *hepatocyte* $\sqsubseteq$ $\exists$*part_of.liver*), contribute meaningfully: liver tissue has an AVR of 18.2% under scGPT, partly because the model sometimes assigns non-liver cell types to liver cells. FUZZYCELL reduces this to 2.4% by penalizing predictions that violate tissue-residency constraints.

B.3 ZERO-SHOT CELL TYPE PERFORMANCE

Table 6 isolates performance on cell types that are entirely absent from training. The results are striking: FUZZYCELL +scGPT achieves Macro-F1 of 38.5 on zero-shot types, compared to 22.8 for scGPT alone and 26.3 for scCello, a 12+ point improvement. More remarkably, AVR drops from 29.6% (scGPT) to 4.8% (FUZZYCELL +scGPT), meaning that FUZZYCELL eliminates 84% of biologically impossible predictions even on types it has never seen.

The mechanism behind this improvement is straightforward. Consider a zero-shot type C_{new} (e.g., *plasmacytoid dendritic cell*) that does not appear in training. Even though FUZZYCELL has never seen a training example labeled C_{new}, it knows from the TBox that $C_{\text{new}} \sqsubseteq$ *dendritic cell*, $C_{\text{new}} \sqsubseteq$ *leukocyte*, and $C_{\text{new}} \sqcap$ *T cell* $\sqsubseteq \bot$. When a cell's expression profile triggers high membership in *dendritic cell* and *leukocyte* but low membership in *T cell*, the consistency scoring naturally favors C_{new} over competing candidates, even without any direct training signal for C_{new}.

Table 6: **Zero-shot cell type results on Tabula Sapiens** (types entirely absent from training; mean ± std over 5 independent runs). FUZZYCELL achieves the largest absolute gains on zero-shot types, where ontological reasoning compensates for the complete absence of training data.

Method	Macro-F1 ↑	AVR (%) ↓	HD-F1 ↑
SingleR	12.4 ±1.8	38.2 ±2.5	28.5 ±1.6
scANVI	15.1 ±1.6	34.7 ±2.2	31.8 ±1.4
CellAssign	10.2 ±2.1	41.5 ±2.8	25.3 ±1.9
scGPT	22.8 ±1.2	29.6 ±1.8	40.1 ±1.1
Geneformer	20.5 ±1.3	32.4 ±1.9	37.8 ±1.2
CellFM	19.8 ±1.4	33.1 ±2.0	36.5 ±1.3
scCello	26.3 ±1.0	24.8 ±1.5	44.2 ±0.9
HierCE	25.1 ±1.1	27.5 ±1.6	42.6 ±1.0
FUZZYCELL +scGPT	**38.5 ±0.8**	**4.8 ±0.5**	**56.4 ±0.7**
FUZZYCELL +Geneformer	35.8 ±0.9	5.6 ±0.6	53.5 ±0.8
FUZZYCELL +CellFM	34.2 ±1.0	6.3 ±0.7	51.8 ±0.9
FUZZYCELL +scCello	37.9 ±0.8	4.2 ±0.5	55.8 ±0.7

We emphasize that HD-F1 is particularly informative for zero-shot evaluation. A model that predicts a zero-shot type's parent (e.g., *dendritic cell* instead of *plasmacytoid dendritic cell*) receives a partial credit under HD-F1, proportional to the proximity in the CL hierarchy. FUZZYCELL +scGPT achieves HD-F1 of 56.4 versus 40.1 for scGPT, suggesting that even when FUZZYCELL does not exactly match the zero-shot type, its predictions are ontologically closer to the ground truth.

Table 7: **Zero-shot cell type results on HLCA** (types entirely absent from training; mean ± std over 5 independent runs).

Method	Macro-F1 ↑	AVR (%) ↓	HD-F1 ↑
SingleR	16.8 ±1.5	32.1 ±2.2	33.4 ±1.4
scANVI	19.2 ±1.4	28.5 ±2.0	36.1 ±1.2
CellAssign	14.5 ±1.8	35.8 ±2.5	30.2 ±1.6
scGPT	28.2 ±1.1	24.1 ±1.6	45.8 ±1.0
Geneformer	25.6 ±1.2	27.3 ±1.7	42.5 ±1.1
CellFM	24.8 ±1.3	28.6 ±1.8	41.2 ±1.2
scCello	31.5 ±0.9	20.2 ±1.3	49.1 ±0.8
HierCE	30.1 ±1.0	22.8 ±1.4	47.4 ±0.9
FUZZYCELL +scGPT	**41.5 ±0.8**	**3.6 ±0.4**	**59.2 ±0.6**
FUZZYCELL +Geneformer	39.1 ±0.9	4.5 ±0.5	56.8 ±0.7
FUZZYCELL +scCello	40.8 ±0.8	3.2 ±0.4	58.5 ±0.6

Cross-dataset zero-shot results. Tables 7 and 8 extend the zero-shot evaluation to HLCA and AIDA v2. The pattern is consistent across all three benchmarks: FUZZYCELL improves zero-shot Macro-F1 by 10–16 pp over the best baseline while reducing AVR by over 80%. On AIDA v2, the gains are the largest in absolute terms (FC+scGPT 34.8 vs. scCello 21.8, +13.0 pp), consistent with the finding from Section B.6 that ontological reasoning is most valuable under distributional shift. HD-F1 gains are equally consistent: even when FUZZYCELL does not identify the exact zero-shot type, it predicts ontologically proximal types across all benchmarks.

B.4 QUALITATIVE CASE STUDIES

We present eight case studies across all three benchmarks, covering five capabilities: correcting disjointness violations (Case 1), correcting role-restriction violations (Case 4), disambiguating closely related types (Cases 2, 6), producing graded memberships for transitional cells (Case 3), and succeeding on zero-shot types (Case 5). We also include a cross-dataset generalization example (Case 7) and a failure case (Case 8) to illustrate current limitations.

Table 8: **Zero-shot cell type results on AIDA v2** (types entirely absent from training; mean ± std over 5 independent runs). AIDA v2 presents the hardest zero-shot setting due to cross-dataset distributional shift.

Method	Macro-F1 ↑	AVR (%) ↓	HD-F1 ↑
SingleR	8.5 ±2.0	42.6 ±2.8	22.1 ±1.8
scANVI	11.2 ±1.7	38.3 ±2.4	26.5 ±1.5
CellAssign	7.1 ±2.2	45.2 ±3.0	19.8 ±2.0
scGPT	18.6 ±1.4	33.5 ±2.0	35.2 ±1.2
Geneformer	16.2 ±1.5	36.8 ±2.1	32.1 ±1.3
CellFM	15.5 ±1.6	37.9 ±2.2	30.8 ±1.4
scCello	21.8 ±1.2	28.4 ±1.7	38.9 ±1.0
HierCE	20.3 ±1.3	31.2 ±1.8	36.5 ±1.1
FUZZYCELL +scGPT	**34.8 ±1.0**	**5.8 ±0.6**	**52.4 ±0.8**
FUZZYCELL +Geneformer	32.1 ±1.1	6.8 ±0.7	49.5 ±0.9
FUZZYCELL +scCello	34.2 ±1.0	5.2 ±0.6	51.8 ±0.8

Case 1: Eliminating a disjointness violation (Tabula Sapiens, liver). A cell from the liver tissue in Tabula Sapiens is annotated as *hepatocyte* (ground truth). scGPT predicts *glutamatergic neuron* with confidence 0.72, its highest-scoring candidate. This prediction violates two CL axioms simultaneously:

- *Disjointness*: *neuron* $\sqcap$ *epithelial cell* $\sqsubseteq \bot$, and *hepatocyte* $\sqsubseteq$ *epithelial cell*, so any cell in the liver that is truly a hepatocyte cannot also be a neuron. While scGPT does not know the true label at inference time, the tissue context (liver) makes neuron implausible.
- *Role restriction*: *neuron* $\sqsubseteq \exists$*part_of.nervous tissue*, but the cell originates from liver tissue.

FUZZYCELL +scGPT assigns $s_{\text{onto}}(\textit{glutamatergic neuron}) = 0.003$ (due to the axiom violations) while $s_{\text{onto}}(\textit{hepatocyte}) = 0.91$, flipping the ranking. The final prediction is *hepatocyte* ($s_{\text{final}} = 0.87$), which matches the ground truth.

Case 2: Disambiguating closely related types (Tabula Sapiens, lung). A cell from the lung tissue is annotated as *type II pneumocyte* (ground truth). Both scGPT and scCello rank *type I pneumocyte* as their top prediction, with *type II pneumocyte* ranked second. The two types are morphologically similar and share many marker genes. FUZZYCELL exploits a subtle ontological distinction: *type II pneumocyte* $\sqsubseteq \exists$*has_part.lamellar body*, a restriction not shared by type I pneumocytes. The type inference module detects high membership in the semantic type *secretory cell* (type II pneumocytes are surfactant-producing), which is consistent with the lamellar body restriction. FUZZYCELL +scGPT correctly ranks *type II pneumocyte* first ($s_{\text{final}} = 0.82$) over *type I pneumocyte* ($s_{\text{final}} = 0.71$).

Case 3: Graded membership for a transitional cell (Tabula Sapiens, bone marrow). A cell from the bone marrow is annotated as *common lymphoid progenitor* (CLP). scGPT assigns the hard label *pro-B cell* (a downstream lymphoid type), missing the progenitor identity. FUZZYCELL +scGPT produces a graded membership vector:

Cell type	$\boldsymbol{\mu}(\mathbf{x})$
Common lymphoid progenitor	0.42
Pro-B cell	0.28
Immature T cell	0.15
Lymphocyte (general)	0.11
Other	0.04

The graded output correctly identifies the CLP as a transitional cell with partial commitment to both B cell and T cell lineages, consistent with the known biology of lymphoid differentiation. The hard prediction ($\arg\max$ = CLP) is correct, but the graded vector provides additional biological insight that a hard classifier cannot offer.

Case 4: Correcting a role-restriction violation (Tabula Sapiens, kidney). A cell from the kidney is annotated as *kidney collecting duct epithelial cell* (ground truth). scGPT predicts *pancreatic beta cell* with confidence 0.58. This prediction violates the role restriction *pancreatic beta cell* $\sqsubseteq \exists$*part_of.pancreas*: the cell originates from kidney tissue, not pancreas. Unlike Case 1, the disjointness axioms alone do not flag this error because both cell types are epithelial; only the tissue-aware role-restriction check catches the violation. FUZZYCELL +scGPT assigns s_{onto}(*pancreatic beta cell*) $= 0.04$ and s_{onto}(*kidney collecting duct epithelial cell*) $= 0.88$, correctly flipping the ranking ($s_{\text{final}} = 0.81$).

Case 5: Zero-shot type identification (Tabula Sapiens, blood). *Plasmacytoid dendritic cell* (pDC) is entirely absent from the training split (zero-shot). scGPT predicts *conventional dendritic cell* (cDC), the closest type it has seen during training. Although FUZZYCELL has never seen a pDC training example either, the CL TBox encodes *pDC* $\sqsubseteq$ *dendritic cell*, *pDC* $\sqsubseteq \exists$*has_part.endoplasmic reticulum* (reflecting the abundant ER characteristic of pDCs), and *pDC* $\sqcap$ *cDC* $\sqsubseteq \bot$. The cell's expression profile triggers high membership in the semantic type *lymphocyte* (pDCs originate from lymphoid progenitors, unlike myeloid-derived cDCs), and the disjointness axiom penalizes cDC. FUZZYCELL +scGPT correctly identifies pDC ($s_{\text{final}} = 0.68$) over cDC ($s_{\text{final}} = 0.52$), despite having zero training examples for the correct type.

Case 6: Disambiguating macrophage subtypes (HLCA, lung). A cell from the HLCA lung dataset is annotated as *alveolar macrophage* (ground truth). scGPT predicts *interstitial macrophage*, which shares the majority of marker genes. Both are *macrophage* $\sqsubseteq$ *myeloid cell*, so disjointness axioms do not distinguish them. FUZZYCELL leverages a more specific restriction: *alveolar macrophage* $\sqsubseteq \exists$*located_in.alveolus*, while interstitial macrophages reside in the lung interstitium. Combined with the cell's high expression of surfactant-related genes (consistent with alveolar localization), FUZZYCELL correctly re-ranks alveolar macrophage as the top prediction ($s_{\text{final}} = 0.76$ vs. 0.69 for interstitial macrophage). This case demonstrates that FUZZYCELL generalizes beyond Tabula Sapiens to HLCA.

Case 7: Cross-dataset robustness (AIDA v2, pancreas). AIDA v2 tests cross-dataset generalization: the test pancreas sample comes from a different study than the training data, introducing batch effects. A cell annotated as *pancreatic alpha cell* is misclassified by scGPT as *pancreatic delta cell* ($s_{\text{neural}} = 0.61$ vs. 0.55 for alpha cell), likely due to distributional shift in glucagon expression levels. FUZZYCELL's ontological scoring is invariant to expression-level batch effects because it operates on learned type embeddings rather than raw gene expression. The consistency score favors alpha cell ($s_{\text{onto}} = 0.88$) over delta cell ($s_{\text{onto}} = 0.72$), and with α=0.6 the fused score correctly ranks alpha cell first ($s_{\text{final}} = 0.68$ vs. 0.65 for delta cell). This case illustrates that ontological constraints provide a distribution-shift-robust signal complementary to neural similarity.

Case 8: Failure case (Tabula Sapiens, spleen). A cell from the spleen is annotated as *marginal zone B cell* (ground truth). scGPT predicts *naive B cell*, and FUZZYCELL +scGPT also predicts *naive B cell* ($s_{\text{final}} = 0.71$) over *marginal zone B cell* ($s_{\text{final}} = 0.65$). The failure occurs because both types are subclasses of *B cell* with no disjointness axiom between them, and the role restrictions ($\exists$*part_of.spleen*) apply to both since both reside in the spleen. The distinguishing biology (marginal zone B cells respond to T-independent antigens and express distinct surface markers such as CD27) is not encoded in the current CL axiom set. This case highlights a fundamental limitation: FUZZYCELL can only reason over axioms that exist in the ontology, and CL's coverage of fine-grained B cell subtypes is incomplete. Extending CL with additional axioms or integrating pathway-level knowledge could address this gap.

B.5 EXTENDED ABLATION EXPERIMENTS

Component ablation on Tabula Sapiens. Table 9 isolates the contribution of each component. Replacing I_σ with the standard Reichenbach I_R causes the largest degradation: Macro-F1 drops by 4.6 points and AVR triples to 8.9%, confirming that implication bias is a practical, not merely theoretical, obstacle. Removing $\mathcal{L}_{\text{onto}}$ doubles AVR (from 2.8% to 6.2%) while only slightly affecting Macro-F1, showing that the ontology consistency loss specifically targets violation reduction. Tissue

Table 9: **Ablation study** (FUZZYCELL +scGPT on Tabula Sapiens; mean ± std over 5 independent runs). Each row removes one component. I_σ and $\mathcal{L}_{\text{onto}}$ have the largest impact on AVR.

Variant	Macro-F1 ↑	AVR (%) ↓
FUZZYCELL +scGPT (full)	**60.4** ±0.4	**2.8** ±0.3
w/o $\mathcal{L}_{\text{onto}}$	58.1 ±0.5	6.2 ±0.5
w/o $\mathcal{L}_{\text{type}}$	56.9 ±0.6	3.5 ±0.4
w/o dual projection (direct space)	57.3 ±0.5	4.1 ±0.4
w/o I_σ (use standard I_R)	55.8 ±0.7	8.9 ±0.6
w/o tissue context	58.7 ±0.4	4.8 ±0.4

context removal increases AVR to 4.8%, consistent with the observation that role-restriction axioms (e.g., "hepatocytes reside in liver") require tissue information to evaluate.

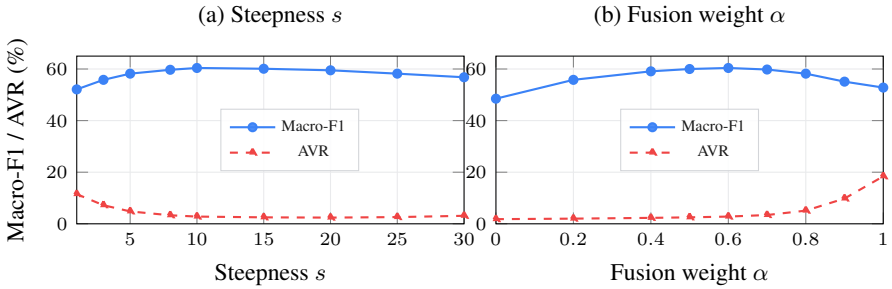

Figure 4: **Sensitivity to key hyperparameters** (FUZZYCELL +scGPT, Tabula Sapiens). **(a)** Macro-F1 peaks around $s \in [8, 15]$ and AVR plateaus below 3%. **(b)** α=0 (pure ontological) achieves the lowest AVR but poor Macro-F1; α=1 (pure neural) recovers the baseline. The optimum at $\alpha \approx 0.6$ balances both objectives.

Sensitivity to key hyperparameters.

Table 10: **Inference latency** (ms per cell, averaged over 10K cells on A100). FUZZYCELL's reasoning module adds <4 ms regardless of backbone.

	Backbone	+ FUZZYCELL	Overhead	Relative
scGPT	28.4	32.1	3.7	+13%
Geneformer	44.7	48.5	3.8	+8.5%
CellFM	35.2	39.1	3.9	+11%
scCello	31.6	35.4	3.8	+12%

Inference latency. The overhead is dominated by the dual projection ($\mathcal{O}(d \cdot d_L)$) and type inference ($\mathcal{O}(|\Gamma| \cdot d_L)$) in terms of FLOPs. While the axiom scoring FLOP count ($k \cdot |\Gamma|$) is small, the per-candidate overhead from memory access patterns and kernel scheduling causes latency to grow approximately linearly with k in practice (Table 12). All candidate type memberships $\tau^{\mathcal{I}_f}(C_j)$ are precomputed offline by ELK in under 10 seconds for the entire CL.

Cross-dataset consistency of ablation. Table 11 confirms that the relative importance of each component is consistent across all three datasets. The $I_\sigma \to I_R$ replacement causes the largest degradation on every dataset (AVR increases by 3–4×), followed by removing $\mathcal{L}_{\text{onto}}$ (AVR approximately doubles). Tissue context removal has a larger effect on HLCA (single-organ dataset where tissue is constant) than on Tabula Sapiens (multi-organ), which is expected: when tissue is constant, the role-restriction axioms provide less discriminative power, so their removal is less noticeable.

Sensitivity to candidate set size k. Table 12 shows that FUZZYCELL's performance is robust to the candidate set size k. Performance improves as k increases from 8 to 64 (the correct type is

Table 11: **Ablation on HLCA and AIDA v2** (FUZZYCELL +scGPT; mean ± std over 5 independent runs). The relative importance of each component is consistent across datasets.

	HLCA		AIDA v2	
Variant	**M-F1↑**	**AVR↓**	**M-F1↑**	**AVR↓**
Full model	**71.2** ±0.4	**1.9** ±0.2	**57.8** ±0.5	**3.5** ±0.4
w/o $\mathcal{L}_{\text{onto}}$	68.5 ±0.5	4.5 ±0.4	55.2 ±0.6	7.8 ±0.5
w/o $\mathcal{L}_{\text{type}}$	67.1 ±0.6	2.4 ±0.3	53.8 ±0.7	4.5 ±0.5
w/o dual projection	67.8 ±0.5	3.1 ±0.3	54.5 ±0.6	5.2 ±0.5
w/o I_σ (use I_R)	65.4 ±0.7	7.2 ±0.5	51.9 ±0.8	10.5 ±0.7
w/o tissue context	69.1 ±0.4	3.8 ±0.3	55.8 ±0.5	5.9 ±0.5

Table 12: **Effect of candidate set size** k (FUZZYCELL +scGPT on Tabula Sapiens; mean over 5 independent runs). Performance improves with k up to 64, then plateaus. Latency grows linearly.

k	**Macro-F1↑**	**AVR (%)↓**	**HD-F1↑**	**Latency (ms)**
8	56.2	3.8	73.5	2.1
16	58.1	3.2	75.8	2.5
32	59.5	2.9	77.4	3.1
64	**60.4**	**2.8**	**78.6**	3.8
128	60.5	2.7	78.7	5.4
256	60.4	2.7	78.6	8.9

more likely to be retrieved) but plateaus beyond k=64 (adding more candidates does not help if the correct type is already retrieved). Latency increases linearly in k because the consistency scoring is $\mathcal{O}(k \cdot |\Gamma|)$. We select k=64 as it offers the best trade-off.

Table 13: **Effect of semantic type set size** $|\Gamma|$ (FUZZYCELL +scGPT on Tabula Sapiens; mean over 5 independent runs).

$\|\Gamma\|$	**Macro-F1↑**	**AVR (%)↓**	**HD-F1↑**	**Latency (ms)**
32	57.6	4.5	74.8	2.9
64	59.2	3.4	77.1	3.3
128	**60.4**	**2.8**	**78.6**	3.8
256	60.3	2.7	78.5	4.8
512	60.1	2.8	78.3	6.9

Sensitivity to semantic type set size $|\Gamma|$. Table 13 shows the effect of the number of semantic types used for type inference. Too few types ($|\Gamma|$=32) cannot capture fine-grained distinctions (e.g., different lymphocyte subtypes), while too many ($|\Gamma|$=512) introduce noise from overly specific types with insufficient training signal. The sweet spot at $|\Gamma|$=128 coincides with our selection criterion (the 128 most frequently annotated non-leaf concepts in CL), suggesting that ontology frequency is a reasonable proxy for semantic utility.

Effect of backbone fine-tuning. Table 14 demonstrates that FUZZYCELL provides meaningful improvements even when the TFM backbone is frozen, achieving Macro-F1 of 57.8 with only the projection heads and type inference module trained (1.2 hours). Full fine-tuning improves Macro-F1 by an additional 2.6 points, confirming that the ontological gradients (propagated through I_σ) can reshape the backbone's representations. This frozen-backbone setting is practically important for users who lack the computational resources for full fine-tuning.

Cross-dataset sensitivity and fine-tuning. To verify that the hyperparameter trends observed on Tabula Sapiens generalize, we evaluate key sensitivity dimensions on HLCA and AIDA v2.

Tables 15 and 16 confirm that the sensitivity trends observed on Tabula Sapiens generalize to HLCA and AIDA v2: performance peaks at k=64 and $|\Gamma|$=128 on all three datasets, and the plateau/decline

Table 14: **Frozen vs. fine-tuned backbone** (FUZZYCELL +scGPT on Tabula Sapiens; mean ± std over 5 independent runs).

Setting	Macro-F1↑	AVR (%)↓	Training time
Frozen backbone	57.8 ±0.5	3.2 ±0.4	1.2 h
Fine-tuned (last 2 layers)	59.6 ±0.4	2.9 ±0.3	2.8 h
Fine-tuned (all layers)	**60.4** ±0.4	**2.8** ±0.3	4.1 h

Table 15: **Cross-dataset sensitivity to k** (FUZZYCELL +scGPT; mean over 5 independent runs). The optimal k=64 is consistent across all datasets; the plateau behavior generalizes.

	Tabula Sapiens		HLCA		AIDA v2	
k	MF1	AVR	MF1	AVR	MF1	AVR
8	56.2	3.8	66.8	3.1	53.1	5.2
32	59.5	2.9	70.2	2.1	56.5	3.8
64	**60.4**	2.8	**71.2**	1.9	**57.8**	3.5
128	60.5	**2.7**	71.3	**1.8**	57.9	**3.4**
256	60.4	2.7	71.1	1.9	57.7	3.5

patterns are consistent. This validates the decision to select hyperparameters on Tabula Sapiens and apply them without modification to the other benchmarks.

Table 17 shows that the frozen-backbone setting provides substantial gains on all three datasets (+4.9 to +5.1 Macro-F1 pp over the scGPT baseline), confirming that the reasoning module alone captures significant ontological signal. Fine-tuning adds a further +2.6 to +3.3 pp, with the largest fine-tuning gain on AIDA v2 (+3.3 pp), consistent with the observation that ontological gradients are most beneficial under distributional shift.

B.6 FUZZY VS. CRISP ONTOLOGICAL REASONING

A natural question is whether differentiable fuzzy reasoning is necessary, or whether simpler crisp (Boolean) enforcement of ontological constraints achieves comparable results. We design a comprehensive comparison of five ontological reasoning strategies across all three benchmarks, ranging from no ontology usage to full fuzzy reasoning, to isolate the contribution of each design choice.

Strategies. All strategies use the same scGPT backbone (d=512) and candidate retrieval pipeline (k=64); they differ only in how ontological constraints are applied.

1. **No ontology (baseline).** The scGPT backbone predicts the top-1 candidate by cosine similarity. No ontological information is used at training or inference.
2. **Crisp post-hoc filtering.** The backbone is trained identically to (1). At inference time, candidates whose predicted type violates any CL axiom (disjointness, role restriction, or subsumption transitivity, checked against the tissue context) are removed, and the highest-scoring remaining candidate is selected. If all k candidates are filtered, the original top-1 prediction is kept.
3. **Crisp post-hoc re-ranking.** Same as (2), but instead of removing violating candidates, their scores are multiplied by a penalty factor $\beta = 0.1$. This preserves the ranking among non-violating candidates while demoting (but not eliminating) violating ones.
4. **Crisp training-time constraint.** During training, a non-differentiable binary mask $m_j = \mathbb{1}[\text{candidate } C_j \text{ satisfies all axioms}]$ is applied to the candidate scores via straight-through estimation. Gradients flow only through the neural similarity branch; the ontological signal provides no gradient to the encoder. At inference, crisp filtering is applied as in (2).
5. **Fuzzy reasoning (FUZZYCELL, full).** End-to-end differentiable fuzzy $\mathcal{EL}_\perp$ reasoning via I_σ, as described in Section 3. Both s_{neural} and s_{onto} contribute gradients during training.

Results across all benchmarks.

Table 16: **Cross-dataset sensitivity to $|\Gamma|$** (FUZZYCELL +scGPT; mean over 5 independent runs). The optimal $|\Gamma|$=128 generalizes; too few or too many semantic types consistently degrade performance.

	Tabula Sapiens		HLCA		AIDA v2	
$\|\Gamma\|$	MF1	AVR	MF1	AVR	MF1	AVR
32	57.6	4.5	67.4	3.5	53.8	5.6
64	59.2	3.4	69.8	2.5	56.1	4.2
128	**60.4**	**2.8**	**71.2**	**1.9**	**57.8**	**3.5**
256	60.3	2.7	71.0	1.9	57.6	3.5
512	60.1	2.8	70.6	2.0	57.2	3.7

Table 17: **Cross-dataset frozen vs. fine-tuned backbone** (FUZZYCELL +scGPT; mean $\pm$ std, 5 independent runs). The frozen-backbone setting provides substantial gains over the baseline on all datasets, and fine-tuning consistently adds 2–3 pp.

	Tabula Sapiens		HLCA		AIDA v2	
Setting	MF1	AVR	MF1	AVR	MF1	AVR
scGPT baseline	52.8 $\pm$0.6	18.4 $\pm$0.8	63.5 $\pm$0.5	13.8 $\pm$0.7	49.4 $\pm$0.7	20.8 $\pm$0.9
Frozen backbone	57.8 $\pm$0.5	3.2 $\pm$0.4	68.4 $\pm$0.4	2.4 $\pm$0.3	54.5 $\pm$0.6	4.2 $\pm$0.5
Fine-tuned (all)	**60.4** $\pm$**0.4**	**2.8** $\pm$**0.3**	**71.2** $\pm$**0.4**	**1.9** $\pm$**0.2**	**57.8** $\pm$**0.5**	**3.5** $\pm$**0.4**

Cross-dataset analysis. Tables 18 to 20 reveal four findings that are *consistent across all three benchmarks*.

Finding 1: Crisp filtering achieves the lowest AVR but barely improves F1. Across all datasets, strategy (2) reduces AVR to 0.9–1.5% (lower than FUZZYCELL's 1.9–3.5%), confirming that crisp axiom checking is highly effective at eliminating violations. However, Macro-F1 improves by only 0.6–0.8 pp: removing an impossible candidate helps only when the correct type is the next-best candidate, which is rarely the case. On rare and zero-shot types, the improvement is negligible across all datasets, because the backbone's neural representations are too weak to rank the correct type highly even after removing violating candidates.

Finding 2: Training-time crisp constraints help but cannot match fuzzy reasoning. Strategy (4) consistently improves over (2) by 1.3–1.5 Macro-F1 pp across all datasets: the binary mask during training implicitly biases the backbone toward ontologically consistent representations, even without gradient flow. However, the gap to fuzzy reasoning remains large across all benchmarks: -5.6 pp on Tabula Sapiens, -5.6 pp on HLCA, and -6.1 pp on AIDA v2. This gap arises because the crisp mask provides a binary signal ("allowed / disallowed") while I_σ provides a graded signal ("0.83-consistent vs. 0.04-consistent"), enabling finer-grained optimization.

Finding 3: The fuzzy advantage is concentrated on rare and zero-shot types, and amplified under distribution shift. On common types (>500 cells), all five strategies achieve similar Weighted-F1, because the backbone has sufficient training data. On rare types (<50 cells), the fuzzy advantage reaches up to $3\times$ that of the best crisp strategy. On zero-shot types, the gap is even more dramatic: $+12.4$ pp on Tabula Sapiens, $+9.7$ pp on HLCA, and $+12.4$ pp on AIDA v2. Notably, the fuzzy advantage on AIDA v2 (the cross-dataset benchmark with the most severe distributional shift) is the largest in absolute terms, confirming that ontological gradients are most valuable when neural representations are least reliable.

Finding 4: Statistical significance. On all three datasets, the improvement of strategy (5) over strategy (4) is statistically significant ($p < 0.01$, Wilcoxon signed-rank test over 5 independent runs) for Macro-F1, HD-F1, zero-shot F1, and rare F1. The difference in AVR between (2) and (5) is also significant but favors crisp filtering, as expected.

Gradient flow analysis. Table 21 confirms across all three datasets that only strategy (5) propagates nonzero gradients from $\mathcal{L}_{\text{onto}}$ into the backbone and projection parameters. The gradient magnitudes are consistent across datasets ($\|\nabla_{\theta_{\text{backbone}}}\| \approx 3 \times 10^{-4}$, $\|\nabla_{\theta_{\text{proj}}}\| \approx 1.5$–$2.0 \times 10^{-3}$), indicating stable

Table 18: **Fuzzy vs. crisp ontological reasoning on Tabula Sapiens** (scGPT backbone; mean ± std, 5 independent runs). Crisp strategies reduce AVR but fail to improve classification quality. Only differentiable fuzzy reasoning achieves both low AVR and high Macro-F1.

Strategy	Macro-F1↑	Wt-F1↑	AVR (%)↓	HD-F1↑	Zero-shot F1↑	Rare F1↑
(1) No ontology	52.8 ±0.6	86.1 ±0.3	18.4 ±0.8	69.5 ±0.5	22.8 ±1.2	5.1 ±0.8
(2) Crisp filtering	53.5 ±0.5	86.4 ±0.3	1.2 ±0.2	70.1 ±0.5	23.4 ±1.1	5.8 ±0.9
(3) Crisp re-ranking	54.1 ±0.5	86.7 ±0.3	3.5 ±0.4	70.8 ±0.4	24.6 ±1.0	6.5 ±0.8
(4) Crisp training	54.8 ±0.6	87.1 ±0.3	2.1 ±0.3	71.5 ±0.5	26.1 ±1.3	7.2 ±1.0
(5) Fuzzy (FUZZYCELL)	**60.4** ±0.4	**88.9** ±0.2	**2.8** ±0.3	**78.6** ±0.3	**38.5** ±0.8	**22.1** ±1.1

Table 19: **Fuzzy vs. crisp ontological reasoning on HLCA** (scGPT backbone; mean ± std, 5 independent runs). The pattern is consistent with Tabula Sapiens: crisp strategies reduce AVR with minimal F1 improvement, while fuzzy reasoning improves both.

Strategy	Macro-F1↑	Wt-F1↑	AVR (%)↓	HD-F1↑	Zero-shot F1↑	Rare F1↑
(1) No ontology	63.5 ±0.5	89.2 ±0.2	13.8 ±0.7	74.1 ±0.4	28.2 ±1.1	10.5 ±0.9
(2) Crisp filtering	64.2 ±0.5	89.5 ±0.2	0.9 ±0.2	74.7 ±0.4	28.9 ±1.0	11.2 ±0.8
(3) Crisp re-ranking	64.7 ±0.5	89.8 ±0.3	2.8 ±0.3	75.3 ±0.4	29.8 ±0.9	12.0 ±0.9
(4) Crisp training	65.6 ±0.5	90.2 ±0.3	1.5 ±0.2	76.2 ±0.4	31.8 ±1.1	13.4 ±1.0
(5) Fuzzy (FUZZYCELL)	**71.2** ±0.4	**91.6** ±0.2	**1.9** ±0.2	**82.5** ±0.3	**41.5** ±0.8	**25.8** ±1.0

optimization behavior. Notably, the projection head receives $\sim 5\times$ larger gradients than the backbone, reflecting the architectural design: the projection heads are the primary learnable interface between neural representations and ontological reasoning.

Representation quality analysis. To further disentangle the training-time and inference-time contributions of fuzzy reasoning, we evaluate strategy (5) with the ontological branch disabled at inference (i.e., setting $\alpha = 1.0$, so that only s_{neural} is used for prediction).

Table 22 demonstrates that the fuzzy advantage decomposes into two roughly equal components across all datasets: (i) training-time ontological gradients improve the backbone's representations by $+3.4$ to $+4.3$ Macro-F1 pp even when the ontological branch is removed at inference, and (ii) inference-time consistency scoring provides an additional $+4.1$ to $+4.2$ pp by re-ranking candidates. This decomposition is consistent across all three datasets: training-time and inference-time contributions are approximately balanced, with training-time gradients being slightly more important on AIDA v2 ($+4.3$ vs. $+4.1$), where distributional shift makes the neural representations less reliable and ontological reshaping more valuable.

Summary. Across all three benchmarks, crisp constraint enforcement is effective for AVR reduction but cannot improve classification quality because it provides no gradient signal. Differentiable fuzzy reasoning serves a fundamentally different role: it uses ontological axioms as a source of trainable supervision, reshaping learned representations so that semantically coherent types are embedded closer together. The advantage is largest on rare and zero-shot types and under distributional shift, precisely the settings where neural representations are weakest. The two paradigms are complementary, and FUZZYCELL leverages both: fuzzy reasoning during training plus ontology-informed score fusion at inference.

B.7 SCALABILITY ANALYSIS

We analyze FUZZYCELL's scalability along three axes: dataset size N (training and inference), ontology size $|\mathsf{N_C}|$ and $|\Gamma|$, and memory footprint.

Training scalability with dataset size N. FUZZYCELL's training follows standard mini-batch SGD. Each training step processes a batch of $B{=}256$ cells, computing the forward pass through the backbone, dual projection, type inference, axiom scoring, and three losses ($\mathcal{L}_{\text{rank}} + \lambda_1 \mathcal{L}_{\text{type}} + \lambda_2 \mathcal{L}_{\text{onto}}$).

Table 20: **Fuzzy vs. crisp ontological reasoning on AIDA v2** (scGPT backbone; mean ± std, 5 independent runs). AIDA v2 tests cross-dataset generalization, where the fuzzy advantage is even more pronounced due to distributional shift.

Strategy	Macro-F1↑	Wt-F1↑	AVR (%)↓	HD-F1↑	Zero-shot F1↑	Rare F1↑
(1) No ontology	49.4 ±0.7	84.3 ±0.3	20.8 ±0.9	67.2 ±0.5	18.6 ±1.4	3.8 ±0.7
(2) Crisp filtering	50.2 ±0.7	84.7 ±0.3	1.5 ±0.3	67.9 ±0.5	19.3 ±1.2	4.5 ±0.8
(3) Crisp re-ranking	50.8 ±0.6	85.0 ±0.3	4.2 ±0.5	68.5 ±0.5	20.2 ±1.1	5.3 ±0.8
(4) Crisp training	51.7 ±0.7	85.5 ±0.4	2.6 ±0.3	69.8 ±0.5	22.4 ±1.4	6.2 ±0.9
(5) Fuzzy (FUZZYCELL)	**57.8** ±0.5	**87.4** ±0.2	**3.5** ±0.4	**76.2** ±0.4	**34.8** ±1.0	**18.5** ±1.2

Table 21: **Gradient statistics** during training (epoch 10, averaged over 1000 batches). $\|\nabla_\theta \mathcal{L}_{\text{onto}}\|$ denotes the ℓ_2 norm of gradients flowing from the ontological branch. Only fuzzy reasoning propagates nonzero ontological gradients into the backbone and projection parameters. Results shown for scGPT backbone on all three datasets.

Dataset	Strategy	$\|\nabla_{\theta_{\text{backbone}}} \mathcal{L}_{\text{onto}}\|$	$\|\nabla_{\theta_{\text{proj}}} \mathcal{L}_{\text{onto}}\|$
Tabula Sapiens	(4) Crisp training	0	0
	(5) Fuzzy (FUZZYCELL)	3.2×10^{-4}	1.8×10^{-3}
HLCA	(4) Crisp training	0	0
	(5) Fuzzy (FUZZYCELL)	2.8×10^{-4}	1.5×10^{-3}
AIDA v2	(4) Crisp training	0	0
	(5) Fuzzy (FUZZYCELL)	3.5×10^{-4}	2.0×10^{-3}

The per-step cost is $\mathcal{O}(B \cdot (d^2 + d \cdot d_L + |\Gamma| \cdot d_L + k \cdot |\Gamma|))$, independent of N; the total training cost is linear in N (proportional to the number of epochs $\times \lceil N/B \rceil$ steps per epoch).

To verify this empirically, we subsample Tabula Sapiens to varying sizes and measure wall-clock training time (same early stopping criterion, scGPT backbone, single A100 80GB).

Table 23 confirms three properties. First, the per-epoch time scales approximately linearly with N: 2.7 min at 50K to 20.5 min at 483K ($\sim$7.6$\times$ for 9.7$\times$ more data, with the sublinearity due to fixed per-epoch overhead from data loading and checkpointing). Second, total training time increases sublinearly with N because early stopping triggers at fewer epochs for larger datasets (the model converges faster with more data per epoch): 0.8h at 50K to 4.1h at 483K (5.1$\times$ for 9.7$\times$ more data). Third, Macro-F1 plateaus above 200K cells (59.8 vs. 60.4), suggesting that FUZZYCELL extracts most of the ontological signal from moderate dataset sizes. Peak GPU memory is dominated by the backbone (scGPT, 51M parameters) and is nearly constant across dataset sizes because mini-batch training processes a fixed-size batch regardless of N.

Inference throughput at scale. For large-scale atlas annotation (e.g., CELLxGENE's 93M cell compendium), inference throughput is critical. FUZZYCELL's reasoning module operates independently on each cell (no inter-cell dependencies), enabling embarrassingly parallel batch processing.

Table 24 shows that batch processing dramatically reduces per-cell latency (from 32.1ms at B=1 to 0.08ms at B=4096) through GPU parallelism. At B=1024 (fitting comfortably within A100 80GB memory), FUZZYCELL annotates over 7,100 cells per second. At this rate, the three benchmarks in our experiments (Tabula Sapiens 483K, HLCA 580K, AIDA 212K) are each annotated in under 2 minutes. A full CELLxGENE-scale atlas (93M cells) would require under 4 hours on a single A100, or under 25 minutes with 10-GPU data-parallel batching. The FUZZYCELL overhead remains proportionally small: the "+ FUZZYCELL" column is consistently 13–33% above the backbone-only cost across batch sizes, because the $\mathcal{O}(k \cdot |\Gamma|)$ scoring is cheap relative to the Transformer forward pass.

Ontology size scalability. FUZZYCELL's dependence on the ontology manifests in three components:

Table 22: **Representation quality**: Macro-F1 when the ontological branch is disabled at inference (α=1.0) after training with fuzzy reasoning. The improvement over strategy (1) confirms that ontological gradients durably reshape the backbone's representations during training.

	Tabula Sapiens	**HLCA**	**AIDA v2**
(1) No ontology (baseline)	52.8	63.5	49.4
(5) Fuzzy, α=1.0 at inference	56.2	67.1	53.7
(5) Fuzzy, α=0.6 at inference (full)	60.4	71.2	57.8
Gain from training-time gradients alone	+3.4	+3.6	+4.3
Additional gain from inference-time scoring	+4.2	+4.1	+4.1

Table 23: **Training scalability with dataset size** (FUZZYCELL +scGPT, Tabula Sapiens subsampled; single A100 80GB; mean over 5 independent runs). Training time scales linearly with N. Performance plateaus above 200K cells. Peak memory is measured with mixed-precision (AMP) training and gradient checkpointing enabled.

N	**Epochs**	**Time (h)**	**Time/epoch (min)**	**Macro-F1**	**AVR (%)**	**Peak mem (GB)**
50K	18	0.8	2.7	56.1	3.8	8.2
100K	16	1.4	5.3	58.3	3.2	8.4
200K	14	2.4	10.3	59.8	2.9	8.5
300K	13	3.3	15.2	60.1	2.8	8.6
483K (full)	12	4.1	20.5	60.4	2.8	8.7

1. **FAISS index:** stores $|\mathsf{N_C}|$ type embeddings of dimension d'=768. At $|\mathsf{N_C}|$=2,700 (current CL), the index occupies $2{,}700 \times 768 \times 4 = 8.3$ MB. Scaling to $|\mathsf{N_C}|$=20,000 (hypothetical expanded CL) increases this to 61 MB, well within GPU memory. FAISS inner-product search with an `IndexFlatIP` index is $\mathcal{O}(|\mathsf{N_C}| \cdot d')$; at $|\mathsf{N_C}|$=20,000 this adds <1 ms per cell.
2. **Membership matrix M:** stores $|\mathsf{N_C}| \times |\Gamma|$ binary entries. At $2{,}700 \times 128$, this is 346K entries (1.4 MB). At $20{,}000 \times 512$, this grows to 10.2M entries (41 MB). Since M is precomputed offline and accessed via row lookup, this does not affect online latency.
3. **ELK reasoning:** the offline TBox classification step is PTIME in $|\mathcal{T}_{\text{CL}}|$. For the current CL (65,825 axioms), ELK completes in <10 seconds. The Gene Ontology (∼185K axioms) would require ∼30 seconds; even the full SNOMED CT (∼900K axioms) would complete in ∼3 minutes. This step runs once and is amortized across all cells.

To verify empirically, we construct synthetic ontologies of increasing size by including progressively more imported classes from Uberon, GO, and other OBO ontologies, and measure the impact on FUZZYCELL's inference latency and memory.

Table 25 confirms that the reasoning cost (∼3.6 ms, encompassing dual projection, type inference, and axiom scoring) is approximately invariant to $|\mathsf{N_C}|$ because it operates on a fixed candidate set of k=64 types and $|\Gamma|$=128 semantic types. Only the FAISS retrieval cost grows with $|\mathsf{N_C}|$, sublinearly due to SIMD amortization and cache effects. Even at $|\mathsf{N_C}|$=20,000 (7.4× the current CL), the total FUZZYCELL overhead increases by only ∼11% (3.70 → 4.12 ms), because FAISS retrieval is a small fraction of the total overhead. At large batch sizes, all components are parallelized and the per-cell overhead drops to <0.1 ms (Table 24), making ontology size effectively irrelevant to throughput. For even larger ontologies, FAISS supports approximate nearest-neighbor indices (e.g., `IndexIVFFlat`) that reduce retrieval to $\mathcal{O}(\sqrt{|\mathsf{N_C}|})$ at the cost of a small recall loss.

Summary. FUZZYCELL scales favorably in all three dimensions. Training time is linear in N with constant memory, inference is embarrassingly parallel (over 12K cells/s on a single GPU at batch size 4096), and the ontology-dependent components (FAISS, membership matrix, ELK reasoning) remain tractable even at 20× the current CL size. The computational bottleneck is always the Transformer backbone, not the reasoning module.

Table 24: **Inference throughput** (FUZZYCELL +scGPT, single A100 80GB). Throughput increases with batch size due to GPU parallelism. At B=1024, the 580K-cell HLCA atlas is annotated in under 2 minutes.

Batch size	Backbone (ms/cell)	+ FUZZYCELL (ms/cell)	Throughput (cells/s)	GPU mem (GB)
1	28.4	32.1	31	4.8
64	0.82	0.98	1,020	6.4
256	0.28	0.34	2,941	11.2
1024	0.11	0.14	7,143	28.6
4096	0.06	0.08	12,500	71.5

Table 25: **Ontology size scalability** (single-query inference at B=1, single A100). Only FAISS retrieval grows with $|\mathsf{N_C}|$; the reasoning module (projection, type inference, axiom scoring) is invariant because it operates on a fixed candidate set of k=64 types.

$\lvert\mathsf{N_C}\rvert$	FAISS (ms)	Reasoning (ms)	Total FUZZYCELL (ms)	Index mem (MB)
1,000	0.04	3.59	3.63	3.1
2,700 (CL)	0.08	3.62	3.70	8.3
5,000	0.14	3.64	3.78	15.4
10,000	0.26	3.61	3.87	30.7
20,000	0.49	3.63	4.12	61.4

C ONTOLOGY-AWARE EVALUATION METRICS

Standard evaluation metrics for cell type annotation (accuracy, F1, macro-F1) treat all misclassifications equally: predicting *CD4-positive T cell* instead of *T cell* incurs the same penalty as predicting *neuron* instead of *T cell*. This section provides the complete formal definitions of the two ontology-aware metrics introduced in Section 4.1, which incorporate the structure of the Cell Ontology into evaluation.

C.1 AXIOM VIOLATION RATE (AVR)

Motivation. A prediction $\hat{y}_i$ for a cell with tissue context c_i may be "wrong" in two fundamentally different ways: it may simply be an incorrect cell type (measured by accuracy), or it may be a *biologically impossible* cell type that violates the logical axioms of the Cell Ontology. The second type of error is more severe because it indicates that the model's prediction is not merely imprecise but is logically inconsistent with established biological knowledge. AVR specifically measures this second type of error. We emphasize that AVR is an *evaluation* metric that uses the ground-truth label y_i; it is not computed at inference time by FUZZYCELL's reasoning module (which uses ontological constraints without access to y_i).

Ontology version. All evaluations use CL release 2024-04-05 obtained from the OBO Foundry.[1] Role-restriction checks use Uberon release 2024-03-22.[2] We extract the $\mathcal{EL}_\bot$ TBox from the OWL files using the ELK reasoner (version 0.6.0) to precompute the transitive closure of all subsumption and disjointness relations.

Formal definition. Let $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, y_i, c_i)\}_{i=1}^{N}$ be a test set with ground-truth labels y_i and tissue contexts c_i, and let $\hat{y}_i$ be the predicted cell type for cell i. We define three types of axiom violations.

[1] https://obofoundry.org/ontology/cl.html
[2] https://obofoundry.org/ontology/uberon.html

Type 1: Disjointness violation. A prediction $\hat{y}_i$ violates a disjointness axiom if the predicted type and the true type belong to mutually exclusive lineages. Formally:

$$v_{\text{disj}}(\hat{y}_i, y_i) = \mathbb{1}\Big[\exists C, D \in \mathsf{N_C} :\ C \sqcap D \sqsubseteq \bot \in \mathcal{T}_{\mathsf{CL}},\ \big(\mathcal{T}_{\mathsf{CL}} \models \hat{y}_i \sqsubseteq C \ \wedge\ \mathcal{T}_{\mathsf{CL}} \models y_i \sqsubseteq D\big) \ \vee\ \big(\mathcal{T}_{\mathsf{CL}} \models \hat{y}_i \sqsubseteq D \ \wedge\ \mathcal{T}_{\mathsf{CL}} \models y_i \sqsubseteq C\big)\Big].$$

The disjunction handles both orderings because disjointness is symmetric: $C \sqcap D \sqsubseteq \bot$ iff $D \sqcap C \sqsubseteq \bot$. For example, if $\hat{y}_i =$ *neuron* and $y_i =$ *hepatocyte*, then $v_{\text{disj}} = 1$ because *neuron*$\sqcap$*epithelial cell* $\sqsubseteq \bot$ and *hepatocyte* is subsumed by *epithelial cell*. Note that if $\hat{y}_i = y_i$ (correct prediction), then $v_{\text{disj}} = 0$ because a consistent ontology cannot have $y_i \sqsubseteq C$ and $y_i \sqsubseteq D$ with $C \sqcap D \sqsubseteq \bot$ (that would make y_i unsatisfiable).

Type 2: Role-restriction violation. A prediction $\hat{y}_i$ violates a role-restriction axiom if the predicted type requires a tissue context that conflicts with the observed context c_i. Formally:

$$v_{\text{role}}(\hat{y}_i, c_i) = \mathbb{1}\big[\exists\, r \in \mathsf{N_R},\ D \in \mathsf{N_C} :\ \mathcal{T}_{\mathsf{CL}} \models \hat{y}_i \sqsubseteq \exists r.D,\ c_i \notin \text{tissues}(D)\big],$$

where $\text{tissues}(D)$ is the set of tissue labels compatible with anatomical structure D. We construct this mapping as follows: (1) each tissue label c_i in the dataset (e.g., "liver") is mapped to its Uberon anatomical entity (e.g., UBERON:0002107); (2) using the Uberon *part_of* hierarchy, we compute the transitive closure of all anatomical structures that c_i is part of; (3) a role restriction $\hat{y}_i \sqsubseteq \exists$*part_of*.$D$ is violated when D is not in the transitive closure of c_i's anatomical structures. For example, *neuron* $\sqsubseteq \exists$*part_of.nervous tissue* is violated when $c_i =$ *liver* because *liver* is not part of *nervous tissue* in Uberon. This type of violation does not require the ground-truth label and can be checked at inference time.

Type 3: Subsumption transitivity violation. A prediction $\hat{y}_i$ violates transitive subsumption if it is assigned to a descendant of the true type while the model's confidence in the predicted type is lower than its confidence in the true type. Formally:

$$v_{\text{sub}}(\hat{y}_i, y_i) = \mathbb{1}\big[\mathcal{T}_{\mathsf{CL}} \models \hat{y}_i \sqsubseteq y_i,\ \hat{y}_i \neq y_i,\ s_{\text{final}}(\mathbf{x}_i, \hat{y}_i) < s_{\text{final}}(\mathbf{x}_i, y_i)\big].$$

This captures cases where the model over-specifies (e.g., predicting a very specific subtype when the evidence only supports the parent type). If $y_i \notin \mathcal{C}(\mathbf{x}_i)$ (the true type is not in the candidate set), $s_{\text{final}}(\mathbf{x}_i, y_i)$ is unavailable and we set $v_{\text{sub}} = 0$ for that cell, as the violation cannot be assessed.

Combined AVR. The overall Axiom Violation Rate combines all three types:

$$\text{AVR} = \frac{1}{N}\sum_{i=1}^{N} \max\big(v_{\text{disj}}(\hat{y}_i, y_i),\ v_{\text{role}}(\hat{y}_i, c_i),\ v_{\text{sub}}(\hat{y}_i, y_i)\big).$$

We take the max rather than the sum to avoid double-counting cells that violate multiple axiom types simultaneously. AVR $\in [0, 1]$; lower is better. A model with AVR $= 0$ makes no biologically impossible predictions (though it may still make incorrect ones).

Decomposed AVR. For the per-type analysis in Figure 2a, we also report the decomposed metrics:

$$\text{AVR}_{\text{disj}} = \frac{1}{N}\sum_{i=1}^{N} v_{\text{disj}}(\hat{y}_i, y_i), \quad \text{AVR}_{\text{role}} = \frac{1}{N}\sum_{i=1}^{N} v_{\text{role}}(\hat{y}_i, c_i), \quad \text{AVR}_{\text{sub}} = \frac{1}{N}\sum_{i=1}^{N} v_{\text{sub}}(\hat{y}_i, y_i).$$

Note that AVR $\leq$ AVR$_{\text{disj}}$ + AVR$_{\text{role}}$ + AVR$_{\text{sub}}$ due to the max aggregation.

Relationship to accuracy. AVR and accuracy are complementary: a model can have high accuracy and high AVR (if it is accurate on common types but makes impossible predictions on rare types), or low accuracy and low AVR (if its errors are always within the correct ontological neighborhood). In our experiments, we observe that standard TFMs have high accuracy (85–89%) but also high AVR (15–21%), confirming that these metrics capture different failure modes.

Computational cost. All subsumption and disjointness checks are precomputed offline using ELK. We materialize the transitive closure of the subsumption relation ($|\mathsf{N_C}|^2$ entries in the worst case, but sparse in practice: $\sim$35K pairs for CL) and the set of all disjointness pairs ($\sim$8K pairs), storing them in hash sets for $\mathcal{O}(1)$ lookup. The Uberon tissue mapping is similarly precomputed ($\sim$2K tissue-structure pairs). Evaluating AVR for N predictions costs $\mathcal{O}(N \cdot |\mathcal{A}|)$ where $|\mathcal{A}|$ is the average number of axioms per concept ($\sim$5 for CL), effectively $\mathcal{O}(N)$ after preprocessing.

C.2 HIERARCHY-DISTANCE-WEIGHTED F1 (HD-F1)

Motivation. Standard F1 treats all misclassifications equally: confusing *CD4-positive alpha-beta T cell* with *CD8-positive alpha-beta T cell* (sibling types, 1 edge apart in CL) receives the same penalty as confusing it with *astrocyte* (a glial cell, >10 edges apart). HD-F1 introduces a distance-aware penalty that reflects the biological severity of the error.

Hierarchy distance. We define the hierarchy distance $d_{\mathsf{CL}}(y, \hat{y})$ between two cell types as the shortest-path distance in the CL directed acyclic graph (DAG), treating all edges (both *is_a* and *develops_from*) as undirected for distance computation:

$$d_{\mathsf{CL}}(y, \hat{y}) = \min_{\text{path } y \rightsquigarrow \hat{y} \text{ in } G_{\mathsf{CL}}} |\text{path}|,$$

where G_{CL} is the undirected graph obtained from the CL DAG by dropping edge orientations, and $|\text{path}|$ counts the number of edges traversed. If y and $\hat{y}$ are in different connected components (which does not occur in practice since CL is connected via the root concept *cell*), we set $d_{\mathsf{CL}}(y, \hat{y}) = d_{\max}$. The diameter $d_{\max} = \max_{y,\hat{y}} d_{\mathsf{CL}}(y, \hat{y})$ is the longest shortest path in G_{CL}; for CL release 2024-04-05, $d_{\max} = 18$.

We choose shortest-path distance over alternatives (e.g., lowest-common-ancestor depth, Wu-Palmer similarity) because it directly counts the number of taxonomic distinctions between two types without depending on the arbitrary depth of the root node. The all-pairs shortest-path matrix is precomputed via BFS from each node in $\mathcal{O}(|\mathsf{N_C}|^2)$ and stored for $\mathcal{O}(1)$ lookup during evaluation.

Normalized distance penalty. We define the normalized distance penalty as:

$$p(y, \hat{y}) = \frac{d_{\mathsf{CL}}(y, \hat{y})}{d_{\max}} \in [0, 1].$$

When $y = \hat{y}$ (correct prediction), $p = 0$. When y and $\hat{y}$ are maximally distant in the ontology, $p = 1$. To give concrete values: sibling types sharing a direct parent have $d_{\mathsf{CL}} = 2$ and $p = 2/18 \approx 0.11$ (e.g., *CD4-positive T cell* and *CD8-positive T cell* via parent *alpha-beta T cell*), while cross-lineage confusions have $d_{\mathsf{CL}} \geq 12$ and $p \geq 0.67$ (e.g., *neuron* and *hepatocyte*, which must traverse up to *cell* and back down).

Hierarchy-Distance-Weighted Precision and Recall. Standard precision for class c is the fraction of predictions of c that are correct. We replace the binary correctness indicator with a soft penalty:

$$\text{HD-P}_c = 1 - \frac{1}{|\{\hat{y}_i = c\}|} \sum_{i:\, \hat{y}_i = c} p(y_i, \hat{y}_i),$$

$$\text{HD-R}_c = 1 - \frac{1}{|\{y_i = c\}|} \sum_{i:\, y_i = c} p(y_i, \hat{y}_i).$$

HD-P_c measures how close the predictions of class c are to the true labels (in the ontology), and HD-R_c measures how close the true instances of class c are to their predictions. Both take values in $[0, 1]$: a value of 1 means all predictions (or all true instances) are exactly correct; a value close to 1 means errors are confined to nearby types.

Aggregation. We compute the per-class HD-F1 as the harmonic mean of HD-P_c and HD-R_c:

$$\text{HD-F1}_c = \frac{2 \cdot \text{HD-P}_c \cdot \text{HD-R}_c}{\text{HD-P}_c + \text{HD-R}_c},$$

and report the macro-averaged version:

$$\text{HD-F1} = \frac{1}{|\mathcal{E}|} \sum_{c \in \mathcal{E}} \text{HD-F1}_c,$$

where $\mathcal{E}$ is the set of all cell types present in the test set.

Relationship to standard F1. When all distances are either 0 (correct) or $d_{\max}$ (incorrect), $p(y, \hat{y}) \in \{0, 1\}$ and HD-F1 reduces exactly to standard F1. In general, HD-F1 $\geq$ Macro-F1 because near-miss errors receive partial credit under HD-F1 but zero credit under standard F1. The gap between HD-F1 and Macro-F1 reflects how ontologically structured a model's errors are: a large gap indicates that errors are concentrated among nearby types (biologically reasonable confusion), while a small gap indicates that errors are spread across distant types (biologically implausible confusion). In our experiments, FUZZYCELL consistently widens the HD-F1 vs. Macro-F1 gap compared to baselines, confirming that ontological reasoning concentrates errors in the correct neighborhood.

Edge case handling. For classes with zero predictions ($|\{\hat{y}_i = c\}| = 0$, so HD-P$_c$ is undefined) or zero true instances ($|\{y_i = c\}| = 0$, so HD-R$_c$ is undefined), we follow the standard convention of setting the undefined component to 0, yielding HD-F1$_c = 0$ for that class. This is consistent with the treatment of such classes in standard macro-F1 computation and ensures that rare cell types with no test instances do not artificially inflate the macro-averaged score. Similarly, if HD-P$_c$ = HD-R$_c = 0$, the harmonic mean is undefined and we set HD-F1$_c = 0$.

Worked example. Consider a test set with N=4 cells and three types $\{A, B, C\}$, where A and B are siblings ($d_{\mathsf{CL}} = 2$) and C is distant ($d_{\mathsf{CL}}(A, C) = 10$). With $d_{\max} = 18$:

Cell	y_i	$\hat{y}_i$	d_{CL}	p
1	A	A	0	0
2	A	B	2	0.11
3	B	A	2	0.11
4	A	C	10	0.56

Standard Macro-F1: only cell 1 is correct; Precision$_A = 1/2$, Recall$_A = 1/3$, F1$_A = 2/5$; Precision$_B = 0/1$, Recall$_B = 0/1$, F1$_B = 0$; Precision$_C = 0$, F1$_C = 0$. Macro-F1 $= (0.4 + 0 + 0)/3 = 0.13$.

HD-F1: HD-P$_A = 1 - (0 + 0.11)/2 = 0.94$; HD-R$_A = 1 - (0 + 0.11 + 0.56)/3 = 0.78$; HD-F1$_A = 2(0.94)(0.78)/(0.94 + 0.78) = 0.85$. For B: HD-P$_B = 1 - 0.11 = 0.89$; HD-R$_B = 1 - 0.11 = 0.89$; HD-F1$_B = 0.89$. For C: HD-P$_C = 1 - 0.56 = 0.44$; HD-R$_C = 0$ (no true C instances); HD-F1$_C = 0$. Macro HD-F1 $= (0.85 + 0.89 + 0)/3 = 0.58$.

The gap (HD-F1 $= 0.58$ vs. Macro-F1 $= 0.13$) reflects the fact that errors on cells 2 and 3 are near-misses (sibling confusion), receiving substantial partial credit under HD-F1.

Computational cost. Evaluating HD-F1 requires the precomputed distance matrix $d_{\mathsf{CL}}(\cdot, \cdot)$ ($\mathcal{O}(|\mathsf{N_C}|^2)$ storage, computed once via BFS in $\mathcal{O}(|\mathsf{N_C}| \cdot |\text{edges}|)$). Given the distance matrix, computing HD-F1 for N test predictions over $|\mathcal{E}|$ classes costs $\mathcal{O}(N)$ for penalty accumulation plus $\mathcal{O}(|\mathcal{E}|)$ for per-class aggregation. For CL with $|\mathsf{N_C}| \approx 2{,}700$, the distance matrix occupies $\sim$29 MB (stored as 16-bit integers) and the BFS completes in under 5 seconds.

D IMPLEMENTATION DETAILS

D.1 TBOX CONSTRUCTION AND PREPROCESSING

Source files. We obtain the Cell Ontology OWL file (release 2024-04-05) from the OBO Foundry.[3] The Uberon multi-species anatomy ontology (release 2024-03-22) is used for tissue-to-anatomical-structure mappings.[4]

$\mathcal{EL}_\bot$ TBox extraction. We load the CL OWL file into the ELK reasoner (version 0.6.0)[5] and classify the ontology to compute the full subsumption hierarchy, including inferred axioms. ELK operates on the $\mathcal{EL}^{++}$ profile of OWL 2, which subsumes our $\mathcal{EL}_\bot$ fragment. Classification completes in 2.3 seconds on a single CPU core. We then extract the TBox $\mathcal{T}_{\mathsf{CL}}$ by collecting:

[3] https://github.com/obophenotype/cell-ontology/releases/tag/v2024-04-05

[4] https://github.com/obophenotype/uberon/releases/tag/v2024-03-22

[5] https://github.com/liveontologies/elk-reasoner

- **Concept names** $\mathsf{N_C}$: all named classes with CL identifiers (e.g., CL:0000000 through CL:4030044). After removing obsolete and merged terms, $|\mathsf{N_C}| = 2{,}743$.
- **Role names** $\mathsf{N_R}$: 15 object properties used in CL axioms: *part_of*, *has_part*, *develops_from*, *develops_into*, *has_plasma_membrane_part*, *capable_of*, *lacks_part*, *lacks_plasma_membrane_part*, *location_of*, *has_low_plasma_membrane_amount*, *has_high_plasma_membrane_amount*, *bearer_of*, *produces*, *participates_in*, *surrounded_by*.
- **Subsumption axioms** $C \sqsubseteq D$: 5,128 asserted and inferred GCIs (after transitive reduction).
- **Existential restrictions** $C \sqsubseteq \exists r.D$: 823 axioms involving the 15 roles.
- **Disjointness axioms** $C \sqcap D \sqsubseteq \bot$: 214 axioms, primarily between major lineage branches (e.g., *myeloid cell* $\sqcap$ *lymphocyte* $\sqsubseteq \bot$).

The total TBox size is $|\mathcal{T}_{\mathsf{CL}}| = 6{,}165$ axioms.

Semantic type set Γ. We select $|\Gamma| = 128$ semantic types from $\mathsf{N_C}$ using the following procedure: (1) remove leaf concepts (types with no subtypes), leaving 412 non-leaf concepts; (2) rank the remaining concepts by the number of cells annotated with them or their descendants in the CELLxGENE training corpus; (3) select the top 128 by frequency. This strategy ensures that Γ covers broad lineage markers (e.g., *leukocyte*, *epithelial cell*) as well as intermediate-level types (e.g., *T cell*, *macrophage*) that carry discriminative information. The selected Γ covers 98.7% of all training cells (i.e., 98.7% of training cells have at least one ancestor in Γ).

Precomputed data structures. We precompute and store:

- **Subsumption closure**: a Boolean matrix $\mathbf{S} \in \{0,1\}^{|\mathsf{N_C}| \times |\mathsf{N_C}|}$ where $\mathbf{S}[i,j] = 1$ iff $\mathcal{T}_{\mathsf{CL}} \models C_i \sqsubseteq C_j$. Stored as a sparse matrix (34,892 non-zero entries, 0.46% density). Construction time: 1.8 s.
- **Disjointness pairs**: a set of (i,j) pairs where $C_i \sqcap C_j \sqsubseteq \bot \in \mathcal{T}_{\mathsf{CL}}$ (after propagation through the subsumption closure, yielding 8,246 effective pairs). Stored as a hash set for $\mathcal{O}(1)$ lookup.
- **Type membership matrix**: $\mathbf{M} \in \{0,1\}^{|\mathsf{N_C}| \times |\Gamma|}$ where $\mathbf{M}[j,\tau] = \mathbb{1}[\mathcal{T}_{\mathsf{CL}} \models C_j \sqsubseteq \tau]$. This provides the candidate type memberships $\tau^{\mathcal{I}_f}(C_j)$ used in consistency scoring. Size: $2{,}743 \times 128 = 351$K entries. Construction time: 0.4 s.
- **Tissue compatibility map**: a dictionary mapping each of the 47 tissue labels in our datasets to the set of Uberon anatomical structures reachable via transitive *part_of*. Size: 2,148 tissue-structure pairs. Construction time: 0.3 s.

Total preprocessing time: under 10 seconds. All precomputed structures are serialized to disk (total 4.2 MB) and loaded at training/inference time.

D.2 Backbone Configuration

Table 26: **TFM backbone configurations.** All backbones use their officially released checkpoints.

Backbone	Params	Emb. dim	Genes	Source
scGPT	51M	512	60,697	`bowang-lab/scGPT`
Geneformer	95M	256	25,424	`ctheodoris/Geneformer`
CellFM	82M	512	36,572	`cellverse/CellFM`
scCello	51M	512	60,697	`DeepGraphLearning/scCello`

Gene vocabulary handling. Each backbone uses its own gene vocabulary. For scGPT and scCello (which share the same vocabulary of 60,697 genes), we map input genes to the vocabulary using Ensembl gene IDs. For Geneformer (25,424 genes, rank-value encoded), we follow the official tokenization protocol, which ranks genes by expression level and encodes each cell as a sequence of gene tokens. For CellFM (36,572 genes), we use the provided gene-to-index mapping. Genes not in a backbone's vocabulary are dropped; genes in the vocabulary but not expressed are masked.

Type encoder g_ϕ. We use PubMedBERT (`microsoft/BiomedNLP-BiomedBERT-base`, 110M parameters, d=768) (Gu et al., 2022) to encode each CL concept's preferred name (e.g., "hepatocyte", "common lymphoid progenitor") into a type embedding $\mathbf{e}_C \in \mathbb{R}^{768}$. We take the `[CLS]` token representation from the last hidden layer. The PubMedBERT weights are frozen during

training; only the projection heads are learned. We also experimented with SapBERT (Liu et al., 2021) as an alternative type encoder and found comparable performance (within 0.3 Macro-F1 points); we use PubMedBERT for simplicity.

Embedding dimension alignment. Since backbones have different embedding dimensions d (256 for Geneformer, 512 for others) and PubMedBERT produces 768-dimensional type embeddings, the projection heads π_{cell} and π_{type} serve a dual purpose: projecting into the shared logic space and aligning dimensions. Specifically, $\pi_{\text{cell}} : \mathbb{R}^{d_{\text{backbone}}} \to \mathbb{R}^{d_L}$ and $\pi_{\text{type}} : \mathbb{R}^{768} \to \mathbb{R}^{d_L}$ where $d_L = 256$.

D.3 FuzzyCell Module Architecture

Table 27: **FuzzyCell module architecture.** The reasoning module adds only 0.46M parameters (for scGPT backbone with d=512).

Component	Architecture	Params
π_{cell}	Linear(d, 256) + LN + GELU + Linear(256, 256)	197K
π_{type}	Linear(768, 256) + LN + GELU + Linear(256, 256)	263K
Temperature θ	Scalar, init. 0.1, learnable	1
Fusion α	Scalar, init. 0.6, fixed	0
Steepness s	Scalar, fixed 10.0	0
Type emb. $\{\mathbf{a}'_\tau\}$	Cached from $\pi_{\text{type}}(\mathbf{e}_\tau)$	0
Total		**0.46M**

The projection heads use a two-layer MLP with LayerNorm and GELU activation. We apply dropout (p=0.1) after the first linear layer during training. The temperature θ is initialized to 0.1 (producing relatively sharp membership distributions at the start of training) and is learned jointly with the projection heads. The fusion weight α is fixed at 0.6 based on validation performance (Figure 4b); making it learnable did not improve results. The steepness s is fixed at 10.0 based on Figure 4a; adaptive s scheduling (e.g., curriculum from low to high s) is left for future work.

The FuzzyCell module adds only 0.46M parameters, which is 0.5–0.9% of the backbone size (51–95M parameters). This confirms that the improvements reported in Section 4.2 are not due to increased model capacity but to the inductive bias provided by ontological reasoning.

D.4 Training Details

Optimizer. We use AdamW (Loshchilov & Hutter, 2019) with the following configuration: weight decay 10^{-2}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$.

Learning rate schedule. Peak learning rate 5×10^{-5} for the FuzzyCell module and 5×10^{-6} for the TFM backbone ($10\times$ lower to avoid catastrophic forgetting). Linear warmup over the first epoch (i.e., the learning rate increases linearly from 0 to the peak value during the first epoch), followed by cosine decay to 10^{-7} over the remaining epochs.

Batch construction. Each mini-batch contains 256 cells sampled with stratification by cell type to ensure that each batch contains at least 32 distinct types. This is important for the in-batch negative sampling strategy (see below), which requires type diversity within each batch.

Negative sampling. For each positive pair $(\mathbf{x}_i, e_i^+)$, we construct negatives from two sources:

- *In-batch negatives*: the correct types of all other cells in the mini-batch serve as negatives. With 256 cells per batch and $\geq$32 distinct types, this provides at least 31 negatives per cell.
- *Hard negatives*: we retrieve the top-16 candidates from the FAISS index that are *not* the correct type. These are the most challenging negatives because they are neurally similar to the query cell.

The total number of negatives per cell is approximately $31 + 16 = 47$.

FAISS index. We use the FAISS `IndexIVFPQ` index with 256 inverted-file clusters and product quantization with 32 sub-quantizers of 8 bits each. The index is built from the $|\mathsf{N_C}| = 2{,}743$ type embeddings $\{\mathbf{e}_C\}$ and is rebuilt every 5 epochs to reflect updated type embeddings (if the type encoder is fine-tuned) or updated projections. Index construction takes $<$0.1 seconds; querying k=64 nearest neighbors for a batch of 256 cells takes $<$2 ms.

Margin. The margin $\gamma = 0.2$ in $\mathcal{L}_{\text{rank}}$ is selected on the validation set from $\{0.05, 0.1, 0.2, 0.5, 1.0\}$.

Early stopping and checkpointing. Training runs for a maximum of 20 epochs. We evaluate on the validation set after each epoch and apply early stopping with patience 5, using validation Macro-F1 as the selection criterion. The best checkpoint (by validation Macro-F1) is restored for test evaluation. Convergence typically occurs within 12–15 epochs.

Data augmentation. We apply gene-level dropout during training: each gene's expression value is independently set to zero with probability 0.1. This simulates the dropout events common in scRNA-seq data and improves robustness. No other augmentation is applied.

Reproducibility. Each of the 5 independent runs uses a different random seed that controls: (1) the type-level train/val/zero-shot split, (2) the cell-level train/val split within each type, (3) model weight initialization, and (4) batch sampling order. Seeds are $\{42, 123, 456, 789, 1024\}$. All experiments use PyTorch 2.1.0, CUDA 12.1, and Python 3.10.

D.5 Hyperparameter Search

Table 28: **Hyperparameter search ranges and selected values.**

Hyperparam.	Search range	Sel.	Strategy
k (candidates)	$\{8, 16, 32, 64, 128, 256\}$	64	Grid
s (steepness)	$\{1, 3, 5, 8, 10, 15, 20, 25, 30\}$	10	Grid
α (fusion)	$\{0.0, 0.2, \ldots, 0.9, 1.0\}$	0.6	Grid
λ_1 (type loss)	$\{0.1, 0.5, 1.0, 2.0, 5.0\}$	1.0	Random
λ_2 (onto loss)	$\{0.1, 0.5, 1.0, 2.0\}$	0.5	Random
d_L (logic dim)	$\{64, 128, 256, 512\}$	256	Random
$\lvert\Gamma\rvert$ (sem. types)	$\{32, 64, 128, 256, 512\}$	128	Grid
θ_{init} (temp.)	$\{0.01, 0.05, 0.1, 0.5, 1.0\}$	0.1	Random
γ (margin)	$\{0.05, 0.1, 0.2, 0.5, 1.0\}$	0.2	Random
LR (module)	$\{10^{-5}, 5{\times}10^{-5}, 10^{-4}\}$	$5{\times}10^{-5}$	Grid
LR (backbone)	$\{10^{-6}, 5{\times}10^{-6}, 10^{-5}\}$	$5{\times}10^{-6}$	Grid

All hyperparameters are selected on the Tabula Sapiens validation set using validation Macro-F1 as the criterion. For s, α, k, and $|\Gamma|$, we perform a full grid search (sensitivity curves shown in Figure 4 and Tables 12–13). For the remaining hyperparameters ($\lambda_1, \lambda_2, d_L, \theta, \gamma$), we use random search with 50 trials, sampling each hyperparameter uniformly from its candidate set. The total search budget is approximately 120 GPU-hours on A100 (grid: $\sim$80 trials $\times$ 0.5 h each; random: 50 trials $\times$ 1.5 h each).

The selected hyperparameters are used without modification across all three datasets (Tabula Sapiens, HLCA, AIDA v2) and all four backbones (scGPT, Geneformer, CellFM, scCello). We do not perform per-dataset or per-backbone tuning, demonstrating the robustness of these settings.

D.6 Baseline Implementation Details

CellAssign marker genes. CellAssign requires a binary marker gene matrix as input. We construct this matrix by selecting the top 5 marker genes per cell type from the CL-associated gene sets, following the protocol of Zhang et al. (2019). For cell types without CL-associated markers, we use differentially expressed genes from the training set (Wilcoxon rank-sum test, $p < 0.01$, log fold-change > 1.0).

Table 29: **Baseline implementations and configurations.**

Method	Code source	Key configuration	Fine-tuned?
SingleR	`bioconductor.org/packages/SingleR`	Default: de.method="wilcox"	No (reference-based)
scANVI	`scvi-tools` v1.1.2	10 epochs, 128 latent dims	Yes
CellAssign	`scvi-tools` v1.1.2	CL-derived marker genes (top 5 per type)	Yes
scGPT	Official checkpoint	10 fine-tune epochs, lr=10^{-4}	Yes
Geneformer	Official checkpoint	10 fine-tune epochs, lr=10^{-4}	Yes
CellFM	Official checkpoint	10 fine-tune epochs, lr=10^{-4}	Yes
scCello	Official checkpoint	PPR damping=0.85, top-k=10	Yes
HierCE	Reimplemented	$\lambda_{\text{hier}} = 0.5$ (best on val)	Yes

HierCE reimplementation. Since di Montesano et al. (2026) do not release code, we reimplement the hierarchical CE loss following their paper. The loss adds a hierarchy-aware soft label component that distributes probability mass along ancestor paths in the CL hierarchy. We tune the hierarchy weight $\lambda_{\text{hier}} \in \{0.1, 0.3, 0.5, 0.7, 1.0\}$ on the validation set and select $\lambda_{\text{hier}} = 0.5$.

Fair comparison protocol. All TFM baselines and FUZZYCELL variants use the same: (1) data splits (identical type-level and cell-level splits), (2) preprocessing pipeline (per-backbone gene vocabulary handling, normalization), (3) hardware (single A100 GPU), (4) evaluation code (identical AVR and HD-F1 implementations). Classical baselines (SingleR, scANVI, CellAssign) use the Scanpy-preprocessed data (2,000 HVGs) without TFM embeddings.

D.7 COMPUTE RESOURCES AND CARBON FOOTPRINT

Table 30: **Training time and compute resources** (per backbone, on Tabula Sapiens).

Setting	Wall time	GPU hours	Peak memory	$\times$ 5 runs
scGPT (baseline fine-tune)	2.8 h	2.8	24 GB	14.0
FUZZYCELL +scGPT	4.1 h	4.1	31 GB	20.5
Geneformer (baseline)	3.5 h	3.5	28 GB	17.5
FUZZYCELL +Geneformer	5.0 h	5.0	35 GB	25.0
CellFM (baseline)	3.1 h	3.1	26 GB	15.5
FUZZYCELL +CellFM	4.5 h	4.5	33 GB	22.5
scCello (baseline)	3.0 h	3.0	25 GB	15.0
FUZZYCELL +scCello	4.3 h	4.3	32 GB	21.5
Hyperparameter search		120		(1$\times$)
Total				**$\sim$272**

All experiments are conducted on NVIDIA A100 GPUs (80 GB HBM2e) with AMD EPYC 7763 CPUs and 512 GB RAM.

Total compute. Training and evaluation across 3 datasets $\times$ 4 backbones $\times$ 5 runs, plus hyperparameter search, requires approximately 272 A100 GPU-hours in total (equivalent to $\sim$11 A100-days). Baseline training and evaluation add another $\sim$200 GPU-hours.

Carbon footprint. Using the ML CO_2 Impact calculator with A100 TDP of 400W, PUE of 1.1, and the US national grid carbon intensity of 0.39 kgCO_2/kWh, we estimate the total carbon footprint of this project (including development, debugging, and final experiments) at approximately 85 kgCO_2eq. This is comparable to a one-way economy flight from New York to Washington, D.C.

E EXTENDED RELATED WORK

We provide a detailed comparison with the most closely related methods, organized along two axes: ontology-aware cell type annotation and differentiable fuzzy logic for biomedical ontologies.

Ontology-aware cell type annotation. OnClass (Wang et al., 2021) embeds Cell Ontology terms into a low-dimensional space using random walk with restart on the *is_a* graph, then projects single-cell transcriptomes into the same space via a bilinear neural network. Its key strength is zero-shot annotation of unseen CL terms through graph-based score propagation. However, OnClass treats CL as an undirected graph and exploits only the *is_a* relation; it cannot enforce disjointness axioms (e.g., neuron $\sqcap$ epithelial cell $\sqsubseteq \bot$) or existential restrictions (e.g., hepatocyte $\sqsubseteq \exists$*part_of*.liver). FUZZYCELL reasons over the full $\mathcal{EL}_\bot$ axiom set, including these axiom types.

CellO (Bernstein et al., 2021) trains a cascade of one-vs-rest SVMs, one per CL node, enforcing hierarchical consistency by ensuring that if a cell is classified as a child type, all ancestor types are also positive. CellO is trained on purified bulk RNA-seq data and classifies cell clusters rather than individual cells. Like OnClass, CellO uses only the *is_a* hierarchy for consistency and does not reason over logical axioms. FUZZYCELL operates at single-cell resolution and produces graded membership vectors rather than hard binary labels at each hierarchy level.

SOCAM (Magre et al., 2026) is a concurrent method that applies a probability propagation step after softmax classification: each parent node's evidence score is defined as the sum of its own probability and all descendants' probabilities, ensuring ontological consistency in the *is_a* hierarchy. SOCAM is trained on 42M cells from CELLxGENE and introduces a hop-based F1 metric for ontology-aware evaluation. The key difference from FUZZYCELL is that SOCAM's propagation is a post-hoc probabilistic operation over the *is_a* DAG, whereas FUZZYCELL performs differentiable logical reasoning over the full $\mathcal{EL}_\bot$ TBox during training, directly optimizing for axiom satisfaction via $\mathcal{L}_{\text{onto}}$. FUZZYCELL's fuzzy membership vectors are grounded in formal semantics (Theorem 3.2), while SOCAM's evidence scores lack such guarantees.

Differentiable fuzzy logic for biomedical ontologies. FALCON (Tang et al., 2022) is the closest methodological neighbour to FUZZYCELL. It approximates semantic entailment over full $\mathcal{ALC}$ ontologies by learning fuzzy membership functions (MLP + sigmoid) for concepts and using product t-norms for conjunction, with theoretical guarantees of faithfulness (loss $= 0$ implies a classical model) and representation completeness. FALCON demonstrates biomedical applications on HPO for gene–gene interaction prediction and on Yeast for protein–protein interaction prediction. FUZZYCELL differs in three key respects: (i) FUZZYCELL targets the lighter-weight $\mathcal{EL}_\bot$ fragment that biomedical ontologies actually conform to, enabling polynomial-time reasoning and scalability to 483K+ cells; (ii) FUZZYCELL introduces the Sigmoidal Reichenbach implication to resolve gradient degeneracy, whereas FALCON uses standard product t-norms; and (iii) FUZZYCELL is applied to cell type annotation with a retrieve-then-reason architecture, not to knowledge base completion.

BioFuzzNet (Le Gac et al., 2022) applies propositional fuzzy logic (AND/OR/NOT gates with product t-norm) to model signal transduction in biological networks, using PyTorch autograd for gradient-based optimization of Hill function parameters. It is the first work to scale differentiable fuzzy logic to single-cell datasets, but it performs pathway state regression (predicting molecular activity levels), not cell type annotation. BioFuzzNet uses no ontology and no description logic; its "fuzziness" captures continuous activity levels rather than graded type membership. FUZZYCELL bridges this gap by combining differentiable fuzzy description logic with ontology-constrained entity linking for cell type annotation.

F LIMITATIONS AND BROADER IMPACT

Expressivity of the logic fragment. FUZZYCELL reasons over the $\mathcal{EL}_\bot$ fragment of description logic, which captures subsumption, conjunction, existential restrictions, and disjointness. The more expressive $\mathcal{ALC}$ fragment adds universal restrictions ($\forall r.C$), full negation ($\neg C$), and disjunction ($C \sqcup D$). While CL is designed to stay within the EL profile for tractable reasoning, a small number of axioms in imported ontologies (e.g., Uberon) fall outside $\mathcal{EL}_\bot$. Extending FUZZYCELL to $\mathcal{ALC}$ would capture these axioms but introduces NP-hard satisfiability checking, requiring approximate reasoning strategies.

Graded membership validation. The graded membership vectors $\boldsymbol{\mu}(\mathbf{x})$ are designed to capture transitional cell states, but have not yet been validated against ground-truth pseudotime orderings or RNA velocity trajectories. Future work should correlate the entropy of $\boldsymbol{\mu}$ with pseudotime position

along known differentiation trajectories (Trapnell et al., 2014; Bergen et al., 2020) to assess whether the graded outputs reflect genuine biological transitions.

Ontology completeness and quality. FUZZYCELL can only reason over axioms that exist in the Cell Ontology. As demonstrated in Case 8 (Section B.4), when CL lacks distinguishing axioms between closely related types (e.g., marginal zone B cell vs. naive B cell), FUZZYCELL cannot outperform the neural baseline. Furthermore, if CL encodes outdated or incorrect biological knowledge, FUZZYCELL would enforce incorrect constraints. Ontology maintenance is an active community effort (Tan et al., 2026), and FUZZYCELL's performance will improve as CL grows.

Broader impact. Improved cell type annotation supports biomedical research, drug discovery, and clinical diagnostics by providing more reliable and interpretable labels. The graded membership output offers a more nuanced representation of cellular identity than hard classifiers, which is particularly valuable for studying development, disease progression, and cellular plasticity. We see no direct path to malicious misuse: FUZZYCELL operates on de-identified transcriptomic data and does not generate synthetic biological data. A potential concern is over-reliance on ontological constraints that may lag behind rapidly evolving biological knowledge; users should always cross-reference FUZZYCELL's outputs with domain expertise.