

UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Distributed Situation Models: Probabilistic Representations of Truth Conditions

Permalink

<https://escholarship.org/uc/item/6tr9s8pz>

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 48(0)

Author

Brasoveanu, Adrian

Publication Date

2026-09-01

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Distributed Situation Models: Probabilistic Representations of Truth Conditions

Adrian Brasoveanu

Department of Linguistics, UC Santa Cruz

Abstract

In formal semantics, the interpretation function $\llbracket \cdot \rrbracket$ maps linguistic expressions “the pencil is in the cup” to set-theoretic objects $\text{in}(\text{pencil}, \text{cup})$ that mirror the compositional structure of the sentence, but do not encode soft inferences like “the pencil is vertical” that human comprehenders automatically make. Distributed situation models (Frank et al., 2009) were proposed as Gestalt / distributed representations of sentence meanings that enable direct probabilistic inference from their representational geometry. We show that previous implementations of distributed situation models systematically underestimate probabilistic inferences because of their sequential sampling of situations (possible worlds). We propose generating the truth-conditional training data using Markov Chain Monte Carlo sampling over a precomputed pool of valid situations, so that soft constraints are correctly evaluated on complete situations rather than partial ones. Training a competitive neural model on 250,000 MCMC samples yields distributed situation models that accurately encode these inferences, providing reliable, rich meaning representations for cognitive models of language comprehension.

Keywords: situation models; mental models; distributed representations; probabilistic inference; world knowledge; language comprehension; MCMC sampling

Introduction: Problem and Informal Solution

In formal semantics, the interpretation function $\llbracket \cdot \rrbracket$ maps linguistic expressions to set-theoretic objects: individuals (type e), truth values (type t), or functions thereof, e.g., predicates of type et or relations of type $e(et)$ (Dowty et al., 1981; Montague, 1973). A sentence like “the pencil is in the cup” maps to a proposition (type t) of the form $\text{in}_{e(et)}(\text{pencil}_e, \text{cup}_e)$, where types of the set/model-theoretic objects are indicated by subscripts. This is a rough semantic representation sufficient for our purposes; we gloss over the specifics of interpreting definite descriptions etc. here. The structure of this sentential meaning mirrors the compositional structure of the sentence—but deriving $\text{vertical}_{et}(\text{pencil}_e)$ from $\text{in}(\text{pencil}, \text{cup})$ requires explicit inference rules; the representation itself is silent on this soft, defeasible implication.

Yet human comprehenders make such inferences automatically. Stanfield and Zwaan (2001) found that participants who heard “the pencil is in the cup” responded faster to pictures of vertically-oriented pencils, even though orientation was never mentioned. Such findings suggest that comprehension constructs *situation models*—analog mental representations that encode both what is explicitly stated *and* what is im-

plied by world knowledge (Johnson-Laird, 1983; van Dijk & Kintsch, 1983; Zwaan & Radvansky, 1998). Unlike propositional representations, situation models preserve structural relations from the described state of affairs, enabling direct inference without explicit inference-rule applications (Johnson-Laird, 2006). Importantly, situation models are not specific to language—the same mechanisms operate when people process visual narratives (Cohn, 2019)—suggesting a domain-general capacity for understanding states of affairs.

The *mental simulation* view of meaning (Frank & Vigliocco, 2011; Zwaan, 2004) captures this insight: understanding a sentence is constructing a situation (mental) model of the described situation, which is tantamount to mentally ‘simulating’ the described situation. Sentence meanings are analog representations whose geometry mirrors the structure of the world, not of language. The payoff is *direct inference*: a situation-model representation of some fact is also a representation of what the fact implies. An analog representation of a pencil in a cup *is* a representation of a vertical pencil—the geometry encodes this soft, probabilistic inference. In symbolic formal semantics, inferring $\text{vertical}(\text{pencil})$ from $\text{in}(\text{pencil}, \text{cup})$ requires a rule $\forall x. \text{in}(x, \text{cup}) \rightarrow \text{vertical}(x)$, or a defeasible version that exceptionally allows for non-stereotypical cups resting horizontally on their side (Asher & Pelletier, 1997; Reiter, 1980). The representations themselves do not encode what they imply, and inference must be performed *over* them by a separate reasoning system. In an analog representation, the situation models for these propositions overlap because situations where pencils are in cups co-occur with situations where pencils are vertical. Computing the conditional probability $P(\text{vertical} \mid \text{in_cup})$ from situation-model geometry yields a high value directly—no inference-rule applications, just immediate ‘arithmetic’ on the representation itself.

Frank et al. (2003) (building on Golden and Rumelhart 1993; Golden et al. 1994) formalized and computationally implemented this view using Self-Organizing Maps (SOMs, Kohonen 1995). Their *distributed situation models* encode a proposition as a vector in a learned “situation space,” where proximity (vector similarity) reflects how often propositions are jointly true across possible worlds. The geometry of the space, not inference rules, encodes what facts imply about one another. Subsequent work (Frank & Vigliocco, 2011; Frank et al., 2009; Venhuizen et al., 2019, 2022) refines the competitive neural learning algorithm, but all implementations

share a methodological limitation: the training data for the distributed situation models consists of situations / possible worlds represented as binary vectors, which are generated by *sequentially* sampling their binary values 1 / 0 (True / False).

In more detail, a situation / possible world is a binary vector that provides a complete specification of all the relevant features, i.e., atomic propositions, of the world. These features / atomic propositions are either True (1) or False (0), hence binary. A complete situation / possible world is constructed by sampling the truth values of these binary features sequentially, *evaluating probabilistic constraints using only the incomplete information provided by binary features sampled so far*. For example, during the sequential construction of a situation, the algorithm might sample the truth value for `win(charlie)` *before* the truth value for `play_manner(charlie, well)`. Therefore, the probabilistic boost that winning is more likely if Charlie plays well, i.e., that sampling 1 for `win(charlie)` is more likely if `play_manner(charlie, well)` is 1, cannot be applied because the truth value of the conditioning proposition is not yet known. This systematically underestimates soft constraints, which are precisely the probabilistic correlations that situation-model representations must encode for direct inference to work.

This issue cannot be avoided by fixing the sampling order, e.g., always sampling `play_manner(charlie, well)` before `win(charlie)`. On one hand, this is not possible because in any reasonably complex world, there will be many interlocking soft constraints, creating cycles in the dependency graph. On the other hand, fixing the sampling order might introduce unwanted sampling-based biases in the generated situations, distorting the probabilistic correlations that distributed situation models should reflect. Therefore, on each sequential sampling run that constructs a complete situation / possible world, the order of sampling the binary features needs to be randomly shuffled. Consequently, the conditioning proposition will be sampled after the outcome half the time (on average), and the relevant soft constraint cannot affect the outcome at the point when the outcome is sampled.

To estimate accurate probabilistic representations of truth-conditional meanings, which is needed for cognitive models of language comprehension, e.g., for predicting reading times and N400 amplitudes (Frank & Vigliocco, 2011; Venhuizen et al., 2019), we propose **Pool MCMC**. We first enumerate the pool of all valid situations, i.e., all the complete situations that satisfy hard constraints, e.g., Charlie cannot both win and lose. We then do Metropolis sampling over this pool to reflect the soft probabilistic constraints. Crucially, the Metropolis acceptance probability is computed on complete situations.

The rest of the paper is structured as follows. We first describe the microworld that provides the truth-conditional data used to estimate distributed situation models. We then empirically confirm the methodological problem: sequential sampling systematically underestimates soft probabilistic constraints, distorting learned situation-model representations. This is followed by a description of our proposed

solution Pool MCMC. We then present the competitive learning algorithm for distributed situation models and show that they faithfully encode both hard ($P(\text{win}|\text{lose}) = 0.00$) and soft ($P(\text{win}|\text{play_well}) = 0.74$) constraints, with situation-model-derived conditional probability estimates correlating at $r = 0.997$ with underlying situation frequencies. Distributed situation models enable direct inference: hard/soft constraints and their consequences are directly readable from the situation models without symbolic reasoning. Finally, we briefly relate our contribution to prior literature and conclude.¹

The Microworld

To show that analog representations can encode both logical structure and graded probabilistic knowledge, we need a domain where ground truth is known. Following Frank et al. (2003, 2009), we use a microworld methodology—a simplified but structurally rich domain where all constraints are explicit and all probabilities computable.

Entities and Propositions. The microworld contains 17 entities (Table 1) and 44 atomic propositions (Table 2). Three characters—Charlie, Heidi, and Sophia—engage in activities: playing games, playing with toys, winning or losing, and being in locations. A *situation* is a complete assignment of truth values to all 44 propositions: a binary vector $\mathbf{s} \in \{0, 1\}^{44}$.

Category	Entities	N
Persons	charlie, heidi, sophia	3
Games	chess, hide-and-seek, soccer	3
Toys	puzzle, ball, doll	3
Locations	bathroom, bedroom, playground, street	4
Play manners	well, badly	2
Win manners	with-ease, with-difficulty	2

Table 1: Entities in the microworld.

Predicate	N
play(person, game)	$3 \times 3 = 9$
play(person, toy)	$3 \times 3 = 9$
win(person), lose(person)	$3 + 3 = 6$
location(person, place)	$3 \times 4 = 12$
play_manner(person, manner)	$3 \times 2 = 6$
win_manner(manner)	2
Total	44

Table 2: Atomic propositions.

Hard Constraints. The microworld enforces 167 hard constraints: logical requirements specifying which proposition combinations can jointly hold (Table 3). For example: a person cannot both win and lose, or soccer can only be played in

¹The repository https://github.com/abrsvn/distributed_situation_models_probabilistic_truth_conditions_CogSci_2026 makes available the Python reimplementations of the competitive neural model and the 250,000 Pool MCMC and 250,000 sequential situation samples needed to reproduce the results in this paper. (Prior implementations, i.e., Frank and Vigliocco 2011; Frank et al. 2009; Venhuizen et al. 2019, 2022, are MATLAB/C/Prolog based.)

the street. All constraints are implemented as formulas in the Z3 SMT theorem prover (de Moura & Bjørner, 2008). Of the $2^{44} \approx 17.6$ trillion possible situations, only 28,676 satisfy all hard constraints: a reduction factor of ~ 613 million.

Category	N	Examples
Game	19	At most one game per person; chess requires exactly 2 players
Toy	33	At most one toy; puzzle is single-player; soccer requires ball
Location	72	Exactly one location per person; soccer only in street
Winning	31	Cannot both win and lose; at most one winner
Manner	12	Play manner requires playing a game
Total	167	

Table 3: Hard constraint categories.

Soft Constraints. The microworld also includes soft, probabilistic tendencies that make some valid situations more likely than others. These defeasible constraints encode the graded world knowledge that situation models must capture. *Specialty games*: everyone has a favorite game they excel at (Charlie $\rightarrow$ chess, Heidi $\rightarrow$ hide-and-seek, Sophia $\rightarrow$ soccer); winning probability increases $1.5\times$ when playing one’s specialty. *Win manner*: winning at one’s specialty makes “with ease” more likely; winning at a non-specialty makes “with difficulty” more likely. *Play manner*: playing well increases winning probability $1.5\times$, playing badly increases losing probability. *Location preferences*: Heidi and Sophia tend to share locations, while Charlie is less likely to share theirs.

These soft constraints create correlations between propositions, that is, exactly the structure that analog representations must encode. Evaluating them requires knowing *all* relevant proposition values: whether Charlie’s win probability should be boosted depends on whether he is playing chess; whether Heidi’s location should be adjusted depends on where Sophia is. This is the heart of the sampling problem for sequential methods: they evaluate soft constraints on partial information.

The Sampling Problem

Learning distributed situation models requires training data drawn from the target distribution $\pi(s)$ defined by the microworld: hard constraints determine validity, soft constraints assign relative probabilities. Crucially, soft-constraint probability adjustments are defined in terms of *complete* situations, e.g., the probability boost for winning when playing well assumes we know whether the person is playing well.

However, all prior work generates situations sequentially. Frank and Vigliocco (2011) describe the method concisely: situations “were sampled by a non-deterministic algorithm that goes through the 44 atomic [propositions] in a random order and sets each value to 0 or 1, according to the atomic [proposition]’s probability *given the truth values of the atomic*

[propositions] so far” (Appendix A.1). The same approach is used in Frank et al. (2009) and Venhuizen et al. (2019, 2022).

Sequential sampling respects hard constraints: as soon as we sample $\text{win}(\text{charlie}) = 1$, we set $\text{lose}(\text{charlie}) = 0$. But it does not respect the distribution $\pi(s)$ defined by the soft constraints because the probability adjustments are evaluated on *partial* information. Consider the play-manner $\rightarrow$ winning soft constraint: Charlie’s probability of winning should be $1.5\times$ higher when he is playing well. If the algorithm samples $\text{win}(\text{charlie})$ *before* $\text{play_manner}(\text{charlie}, \text{well})$, the adjustment is not applied since the truth value of the conditioning proposition is not yet known. On average, the conditioning proposition will be sampled after the outcome half the time, and the relevant soft constraint cannot affect the outcome at the point when the outcome is sampled.

Empirical Comparison. To quantify this effect, we reimplemented sequential sampling using the same probability adjustments and Z3 constraint checking as Pool MCMC and generated 250,000 samples using both sampling methods. As discussed in the next section, Pool MCMC samples from the target distribution $\pi(s)$ by computing Metropolis acceptance probabilities on complete situations, guaranteeing that soft constraints are evaluated with full information. Table 4 compares conditional probability estimates from each method.

Conditional	Pool	Seq.	Δ
$P(\text{win} \text{play_well})$ Charlie	0.76	0.52	+0.24
$P(\text{win} \text{play_well})$ Heidi	0.93	0.53	+0.40
$P(\text{win} \text{play_well})$ Sophia	0.90	0.53	+0.37
$P(\text{lose} \text{play_badly})$ Charlie	0.97	0.58	+0.38
$P(\text{win} \text{specialty})$ Charlie	0.89	0.46	+0.43
$P(\text{win} \text{specialty})$ Heidi	0.89	0.43	+0.47
$P(\text{win} \text{specialty})$ Sophia	0.84	0.43	+0.41
$P(\text{win} \text{lose})$	0.00	0.00	0.00
$P(\text{street} \text{soccer})$	1.00	1.00	0.00
$P(\text{ball} \text{soccer})$	1.00	1.00	0.00

Table 4: Cond. prob.s estimated from Pool MCMC vs. seq. samples. Soft constraints (top) show large discrepancies; hard constraints (bottom, not dependent on sampling order) agree.

The discrepancies are substantial. Sequential sampling estimates $P(\text{win}|\text{play_well}) \approx 0.53$, barely above the base probability (set at 0.4), while Pool MCMC gives 0.76–0.93. For specialty games, the difference exceeds 40%. Hard constraints show no discrepancy as their enforcement is not sampling-order dependent. Across all 1,892 (44×43) pairwise conditional probabilities, the correlation between the two methods is only $r = 0.75$, with mean absolute difference 0.13 and maximum difference 0.59. Grouping these conditionals by their consequent proposition, the effect is largest for propositions at the end of soft constraint chains: win/lose conditionals show mean $|\Delta| = 0.19$, win manner shows $|\Delta| = 0.27$, while toys (minimally involved in soft constraints) show only $|\Delta| = 0.07$.

Even marginal probabilities differ. Sequential sampling estimates $P(\text{win_manner}(\text{with_ease})) = 0.25$; Pool MCMC gives 0.65. Winning with ease depends on a chain of soft

constraints (playing one’s specialty, playing well, winning) that are underestimated by sequential sampling.

These discrepancies propagate to learned representations. If training samples underestimate the correlation between playing well and winning, the vectors learned for `play_manner(charlie, well)` and `win(charlie)` will have less overlap than they should. The conditional probability $P(\text{win}|\text{play_well})$ computed from representational geometry will be too low, and direct inference, i.e., reading off what one fact implies about another, will yield incorrect answers.

Pool MCMC

We propose Pool MCMC, a two-stage sampling method that generates samples from the target distribution $\pi(\mathbf{s})$ defined by the microworld’s soft constraints. The first stage enumerates all valid situations satisfying hard constraints. The second stage samples from this pool via Metropolis sampling, with acceptance probability computed on *complete* situations.

Stage 1: Enumerating Valid-Situation Pool. The 167 hard constraints reduce the situation space from 2^{44} to a manageable size. Each constraint is encoded as a Z3 formula. Given a partial assignment of truth values to some of the 44 atomic propositions, the Z3 solver determines whether the assignment is consistent with the hard constraints, and whether any unassigned propositions are *forced* to be true or false by the hard constraints (a.k.a., constraint propagation).

We use this capability to enumerate all valid situations by breadth-first search. We seed the search with partially specified situations that assign locations and games for all three persons. Each person must be in exactly one of 4 locations, and plays at most one of 3 games (or none), yielding $4^3 \times 4^3 = 4,096$ seed configurations. For each seed, we apply Z3 constraint propagation to infer additional forced truth values, which are logically determined by constraints given current truth values. We then systematically explore remaining unknown propositions using a queue: for each unknown, we branch on both values (1 and 0; breadth-first search), apply inference closure to determine any newly forced values, and prune branches that lead to contradiction. The enumeration yields **28,676 valid situations**: the complete pool (computed once, reused across all sampling runs) satisfying all 167 hard constraints.²

Goodman and Lassiter (2015) observe that in Church-style probabilistic programming, we do not generally need to construct complete possible worlds. In the cases they consider, the unsampled parts of a situation do not constrain or reweight the sampled parts (for the query at hand), so marginalizing over them is equivalent to ignoring them. This is not the case here: a partial situation cannot be scored by multiplying the factors for the propositions sampled so far, for the same reason sequential sampling leads to underestimation of soft constraints. Correctly scoring a partial situation would require computing and summing the scores of its complete valid extensions.

²Alternatively, an approximate pool can be generated using sequential sampling—appropriate for finding hard-constraint-valid situations (but not for sampling from the target distribution $\pi(\mathbf{s})$).

Stage 2: Metropolis Sampling. We use Metropolis sampling to sample from the pool in accordance to the target distribution $\pi(\mathbf{s}) \propto \prod_{i=1}^{44} p_i(\mathbf{s})^{s_i} (1 - p_i(\mathbf{s}))^{1-s_i}$ —where $p_i(\mathbf{s})$ is the unnormalized, soft-constraint-adjusted probability of proposition $i = 1 \dots 44$ given the complete situation $\mathbf{s}$. If i is true ($s_i = 1$), the contribution is p_i ; if false ($s_i = 0$), $1 - p_i$. Crucially, the soft-constraint adjustments are applied with full information: when computing Charlie’s winning probability, we know whether he is playing chess or playing well.

Proposals. To propose a move from situation $\mathbf{s}$ to $\mathbf{s}'$, we use an equivalence-class approach: (i) select a random subset of K proposition indices to hold fixed ($K = 29$ was tuned for chain mixing efficiency); (ii) find the *equivalence class* of $\mathbf{s}$: all pool situations matching $\mathbf{s}$ on the K indices; (iii) sample uniformly from the equivalence class (excluding $\mathbf{s}$).

This proposal is symmetric: if $\mathbf{s}$ and $\mathbf{s}'$ are part of the same equivalence class (share the same truth values on the K fixed positions), each is equally likely to be proposed from the other, so no Hastings correction is needed. We **accept** the proposal with probability $\alpha = \min\left(1, \frac{\pi(\mathbf{s}')}{\pi(\mathbf{s})}\right)$. We run 20 parallel chains for 60,000 steps each. The first 10,000 steps are discarded as burn-in; from the remaining 1,000,000 samples, we thin by a factor of 4, yielding **250,000 samples** from $\pi(\mathbf{s})$.

Learning Distributed Situation Models

The 250,000 Pool MCMC samples provide training data drawn from the target distribution $\pi(\mathbf{s})$. We now learn distributed situation models from these samples, which are vectorial representations that encode the probabilistic structure of the microworld directly in their geometry.

Competitive Neural Model. Following Frank et al. (2009), we use a competitive neural model with $n = 150$ units / neurons. Each unit $i = 1 \dots n$ maintains a weight vector $\mathbf{w}_i \in [0, 1]^{44}$ initialized at **0.5**, and a bias b_i initialized at 1.0. For each situation $\mathbf{s} \in \{0, 1\}^{44}$ in our 250k training samples:

1. **Compute distances.** $d_i = \sum_{j=1}^{44} |\mathbf{w}_{ij} - s_j|$. L1 / city-block distance sums the degree of mismatch between the unit’s weight vector and the sampled truth-value vector (not the squared mismatch, as in L2 / Euclidean distance).
2. **Select winner.** Winner / Best Matching Unit is unit with smallest bias-adjusted distance: $i^* = \arg \min_i (d_i - b_i)$.
3. **Update winner $\mathbf{w}_{i^*}$ toward input:** $\mathbf{w}_{i^*} \leftarrow \mathbf{w}_{i^*} + \eta_w (\mathbf{s} - \mathbf{w}_{i^*})$.
4. **Update biases.** Winner i^* : $b_{i^*} \leftarrow b_{i^*} + \eta_b \cdot b_{i^*} (1 - b_{i^*})$. Other units i : $b_i \leftarrow b_i + \eta_b \cdot b_i$.

Bias update uses a squashing function for the winner, preventing any unit from dominating. Non-winners receive linear increases, ensuring all units eventually win. Learning rates η_w (for weights) and η_b (for biases) decay linearly over the first third of training ($\eta_w : 0.3 \rightarrow 0.05$; $\eta_b : 10^{-4} \rightarrow 10^{-5}$), then remain constant. We train for 30 epochs, with data randomly shuffled each epoch. The learned probabilities below are computed from the weights only (biases are discarded).

Why the Competitive Model Works. The competitive model learns a compressed representation of the probability-weighted pool of valid situations. Each unit i comes to repre-

sent a cluster of similar situations: its weight vector converges to the mean of the set of situations for which unit i is the winner. That is, each unit i becomes a situation prototype, and the bias mechanism ensures approximately uniform cluster size per prototype. The result is a weight matrix $\mathbf{W} \in [0, 1]^{150 \times 44}$, where each row $\mathbf{W}_i$. ($i = 1 \dots 150$) is a 44-dimensional prototype situation. If instead we look at each of the 44 columns $\mathbf{W}_{\cdot p}$ ($p = 1 \dots 44$), we obtain a *distributed situation model*, i.e., a 150-dimensional vector that is the meaning representation for atomic proposition p . Thus, $\llbracket p \rrbracket = \mathbf{W}_{\cdot p}$. **Figure 1** shows some of these 150-dimensional meanings on the right, and a snippet of the MCMC binary training data on the left.

Meanings for complex propositions can be computed via elementwise operations on the atomic vectors: $\llbracket \neg p \rrbracket = \mathbf{1} - \llbracket p \rrbracket$, $\llbracket p \wedge q \rrbracket = \llbracket p \rrbracket \odot \llbracket q \rrbracket$, and $\llbracket p \vee q \rrbracket = \llbracket p \rrbracket + \llbracket q \rrbracket - \llbracket p \rrbracket \odot \llbracket q \rrbracket$.

The distributed situation models encode accurate probability estimates for priors $P(p) \approx \frac{1}{n} \sum_{i=1}^n \llbracket p \rrbracket_i$, conjunctions $P(p \wedge q) \approx \frac{1}{n} \sum_{i=1}^n \llbracket p \rrbracket_i \cdot \llbracket q \rrbracket_i$, and conditionals $P(p | q) = \frac{P(p \wedge q)}{P(q)}$, where $\llbracket p \rrbracket$, $\llbracket q \rrbracket$ are the meanings / distributed situation models of propositions p, q (vectors with $n = 150$ dimensions). Thus, correct situation sampling is essential since the competitive model learns conditional probabilities from these binary training samples. If they do not reflect the target distribution $\pi(\mathbf{s})$, as with sequential sampling, the learned weights encode incorrect conditionals, and probability estimates derived from them are biased.³

Verification. We verify that the learned distributed situation models faithfully encode the probabilistic structure of the Pool MCMC samples by comparing probability estimates from the 150-dimensional learned vectors against frequencies computed directly from the MCMC samples:

Probability Type	Correlation (r)
Prior $P(p)$	1.000
Conjunction $P(p \wedge q)$	1.000
Conditional $P(p q)$	0.997

Table 5: Correlation between probability estimates from learned vectors and frequencies from Pool MCMC samples.

The near-perfect correlations confirm that the competitive neural model faithfully encodes the distributional structure of its binary training data. Combined with Pool MCMC sampling, the learning procedure produces representations where probability estimates reflect the target distribution $\pi(\mathbf{s})$.

Direct Inference from Representation Geometry

The theoretical payoff of analog situation-model representations is direct inference: inferential relations are encoded in geometry, so representing one fact *is* representing what it implies. We demonstrate that the learned distributed situation

³The competitive model is only one of several ways to compress situation samples into dense probabilistic representations. Irrespective of the final representational format, Pool MCMC will ensure that the underlying samples faithfully reflect the target distribution $\pi(\mathbf{s})$.

models exhibit this property: hard and soft constraints are directly readable from the vectors without symbolic reasoning.

Hard Constraints. Hard constraints manifest as extreme conditional probabilities $P(p | q)$ computed from the learned situation-model vectors, as shown in Table 6. The vectors of two mutually exclusive propositions occupy non-overlapping regions of situation-model space, so conditioning on one yields near-zero probability for the other. But when one proposition implies another, their vectors overlap almost completely.

Conditional	P
<i>Incompatibility ($P \approx 0$)</i>	
$P(\text{win}(\text{charlie}) \text{lose}(\text{charlie}))$	0.00
$P(\text{play}(\text{charlie}, \text{chess}) \text{play}(\text{charlie}, \text{soccer}))$	0.00
<i>Implication ($P \approx 1$)</i>	
$P(\text{location}(\text{charlie}, \text{street}) \text{play}(\text{charlie}, \text{soccer}))$	1.00
$P(\text{play}(\text{heidi}, \text{ball}) \text{play}(\text{heidi}, \text{soccer}))$	0.99

Table 6: Hard constraints yield extreme conditional probabilities based on the learned situation-model vectors.

These values are not stored explicitly. They emerge from geometry: the vectors for $\text{win}(\text{charlie})$ and $\text{lose}(\text{charlie})$ have minimal overlap because situations where Charlie wins never co-occur with situations where Charlie loses; the hard constraint $\neg(\text{win}(\text{person}) \wedge \text{lose}(\text{person}))$ ensures this.

Soft Constraints. Soft constraints manifest as elevated but sub-certain conditional probabilities: the graded world knowledge that situation models intrinsically encode. Table 7 shows conditionals computed from the learned vectors.

Conditional	$P(p q)$	Prior $P(p)$
$P(\text{win}(\text{C}) \text{play_manner}(\text{C}, \text{well}))$	0.74	0.08
$P(\text{lose}(\text{C}) \text{play_manner}(\text{C}, \text{badly}))$	0.96	0.40
$P(\text{win}(\text{with_ease}) \text{win}(\text{C}) \wedge \text{play}(\text{C}, \text{chess}))$	0.76	0.66

Table 7: Soft constraints yield elevated conditional probabilities $P(p|q)$. Prior probabilities $P(p)$ shown for comparison.

Playing well increases Charlie’s winning probability from 0.08 to 0.74, nearly a tenfold increase. Playing badly makes losing highly likely (0.96 vs. prior of 0.40). Winning at one’s specialty game elevates the probability of winning with ease (0.76 vs. 0.66). The third conditional in Table 7 conditions on a conjunction (Charlie both winning and playing chess), demonstrating that direct inference extends to complex conditions (conjunction $\llbracket p \wedge q \rrbracket$ is element-wise product $\llbracket p \rrbracket \odot \llbracket q \rrbracket$). Thus, graded relationships, e.g., the 1.5× probability adjustment factors, are captured in representational geometry.

Figure 1 illustrates the relationship between training data and learned representations. Binary Pool MCMC samples (left) train the competitive neural model, yielding continuous 150-dimensional vectors (right). Brackets show conditional probabilities computed directly from these vectors: hard incompatibilities in red ($P \approx 0$), hard implications in green ($P \approx 1$), soft correlations in blue (elevated but sub-certain).

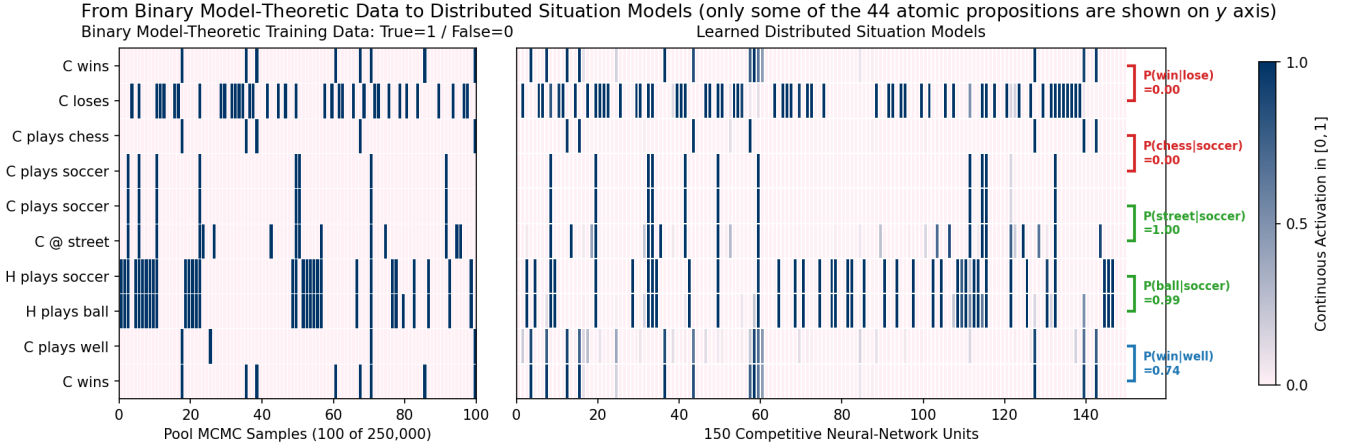


Figure 1: From Pool MCMC to distributed situation models. **Left:** A snippet of the binary 1/0 training data—10 displayed rows, corresponding to 8 unique propositions out of 44 (‘C wins’ and ‘C plays soccer’ are repeated), and 100 Pool MCMC samples (thin columns) out of 250,000. **Right:** Competitively learned 150-dim continuous vectors (distributed situation models) for the same 8 propositions (with the same 2 repeats). Brackets: conditional probabilities computed directly from vector geometry: red = hard incompatibility ($P \approx 0$), green = hard implication ($P \approx 1$), blue = soft correlation (elevated but sub-certain probability).

This is direct inference in action. Winning if playing well is not a result of defeasible inference rules applied to symbolic representations; $P(\text{win}|\text{play_well}) = 0.74$ is directly read off the representational geometry: the overlap between the `win(charlie)` and `play_manner(charlie,well)` vectors in Figure 1, bottom right. The geometry *is* the inference.

Related Work

Mental Models and Situation Models. This work is part of the broader mental models tradition (Johnson-Laird, 1983; Johnson-Laird & Byrne, 1991). Mental models are analog representations whose structure mirrors the structure of what they represent, and reasoning proceeds by inspecting and manipulating the mental model. Situation models (van Dijk & Kintsch, 1983; Zwaan & Radvansky, 1998) extend this to discourse comprehension: understanding a text means constructing a mental model of the described situation. Distributed situation models (Frank et al., 2003, 2009; Venhuizen et al., 2019) operationalize this computationally, learning vector representations where geometric relations encode situational relations. Our contribution ensures these representations are learned from correctly sampled truth-conditional training data.

Grounded vs. Distributional Meaning. Distributed situation models differ from text-derived embeddings. Models like Word2Vec (Mikolov et al., 2013) and BERT (Devlin et al., 2019) learn representations where similarity reflects co-occurrence in corpora: distributional patterns in language use (see Baroni & Lenci, 2010; Boleda, 2020, for surveys). Our representations encode co-occurrence in *possible worlds*: similarity reflects how often propositions are jointly true across situations. This grounds meaning in truth conditions (Montague, 1973) rather than usage patterns. The sampling problem we address is specific to meanings *grounded* in

probabilistic world—not textual—knowledge.

Generics and Typicality. The soft constraints in our microworld—playing well correlates with winning, specialists tend to win their preferred games—are generic / typical generalizations that hold by default but not universally. Our distributed representations intrinsically encode these probabilistic defaults, which are frequent patterns in situation space. With correct situation sampling in place, we can systematically investigate the connections between distributed situation models and formal symbolic accounts of defeasible inference, generics and typicality (Asher & Pelletier, 1997; Reiter, 1980).

Conclusion

Distributed situation models encode sentence meanings as vectors where similarity reflects co-occurrence probability in possible worlds, enabling direct inference: reading off what one fact implies about another from representational geometry. The quality of these representations depends critically on training data that faithfully reflects the probabilistic structure of the world model. We identified a methodological problem shared by all prior implementations of situation models: sequential situation sampling evaluates soft constraints on partial information, systematically underestimating the probabilistic correlations that situation models are meant to encode. We proposed Pool MCMC as a solution: enumerate all valid situations satisfying hard constraints, then sample via Metropolis with acceptance computed on complete situations, ensuring soft constraints are correctly evaluated. Training a competitive neural model on 250,000 Pool MCMC samples yields distributed situation-model representations that faithfully encode in their geometry both hard and soft constraints. Evaluating distributed situation models in downstream models, e.g., predicting RTs or N400, is left for a future occasion.

Acknowledgments

I want to thank the anonymous CogSci 2026 reviewers for their helpful feedback, Harm Brouwer, Stefan Frank, Noortje Venhuizen, and Taylor Webb for conversations related to the topic of this paper, and both Jakub Dotlačil and Awni Altabaa in particular for extended conversations about distributed situation models as representations for truth conditions and related topics. I am grateful to Stefan Frank for sharing his original MATLAB/Prolog implementation of the competitive neural model and sequential situation sampling used in the Frank et al. (2009) paper, which was the starting point for the Python implementation underlying the results in this paper.

LLM assistants were used in circumscribed, closely supervised ways during code development and the editing process of the manuscript. On the spectrum between local, narrow autocomplete vs. global, large-scale LLM-based generation of code or text, this work stays decidedly close to the local / narrow end of the spectrum. I retain full responsibility for all content, and all errors are my own.

References

- Asher, N., & Pelletier, F. J. (1997). Generics and defaults. In J. van Benthem & A. ter Meulen (Eds.), *Handbook of logic and language* (pp. 1125–1177). Elsevier.
- Baroni, M., & Lenci, A. (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4), 673–721.
- Boleda, G. (2020). Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6, 213–234. <https://doi.org/10.1146/annurev-linguistics-011619-030303>
- Cohn, N. (2019). Visual narratives and the mind: Comprehension, cognition, and learning. In K. D. Federmeier & D. M. Beck (Eds.), *Psychology of learning and motivation* (pp. 97–128, Vol. 70). Academic Press.
- de Moura, L., & Bjørner, N. (2008). Z3: An efficient SMT solver. *Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, 337–340.
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 4171–4186. <https://aclanthology.org/N19-1423/>
- Dowty, D. R., Wall, R. E., & Peters, S. (1981). *Introduction to Montague semantics*. Reidel.
- Frank, S. L., Haselager, W. F. G., & van Rooij, I. (2009). Connectionist semantic systematicity. *Cognition*, 110(3), 358–379. <https://doi.org/10.1016/j.cognition.2008.11.013>
- Frank, S. L., Koppen, M., Noordman, L. G. M., & Vonk, W. (2003). Modeling knowledge-based inferences in story comprehension. *Cognitive Science*, 27(6), 875–910. https://doi.org/10.1207/s15516709cog2706_3
- Frank, S. L., & Vigliocco, G. (2011). Sentence comprehension as mental simulation: An information-theoretic perspective. *Information*, 2(4), 672–696. <https://doi.org/10.3390/info2040672>
- Golden, R. M., & Rumelhart, D. E. (1993). A parallel distributed processing model of story comprehension and recall. *Discourse Processes*, 16(3), 203–237.
- Golden, R. M., Rumelhart, D. E., Strickland, J., & Ting, A. (1994). Markov random fields for text comprehension. In D. S. Levine & M. Aparicio IV (Eds.), *Neural networks for knowledge representation and inference* (pp. 283–309). Lawrence Erlbaum Associates.
- Goodman, N. D., & Lassiter, D. (2015). Probabilistic semantics and pragmatics: Uncertainty in language and thought. In S. Lappin & C. Fox (Eds.), *The handbook of contemporary semantic theory* (2nd ed., pp. 655–686). John Wiley & Sons, Ltd. <https://doi.org/10.1002/9781118882139.ch21>
- Johnson-Laird, P. N. (1983). *Mental models: Towards a cognitive science of language, inference, and consciousness*. Harvard University Press.
- Johnson-Laird, P. N. (2006). *How we reason*. Oxford University Press.
- Johnson-Laird, P. N., & Byrne, R. M. J. (1991). *Deduction*. Lawrence Erlbaum Associates.
- Kohonen, T. (1995). *Self-organizing maps* (Vol. 30). Springer.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*. <https://arxiv.org/abs/1301.3781>
- Montague, R. (1973). The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, & P. Suppes (Eds.), *Approaches to natural language: Proceedings of the 1970 stanford workshop on grammar and semantics* (pp. 221–242). Reidel.
- Reiter, R. (1980). A logic for default reasoning. *Artificial Intelligence*, 13, 81–137.
- Stanfield, R. A., & Zwaan, R. A. (2001). The effect of implied orientation derived from verbal context on picture recognition. *Psychological Science*, 12(2), 153–156. <https://doi.org/10.1111/1467-9280.00326>
- van Dijk, T. A., & Kintsch, W. (1983). *Strategies of discourse comprehension*. Academic Press.
- Venhuizen, N. J., Crocker, M. W., & Brouwer, H. (2019). Semantic entropy in language comprehension. *Entropy*, 21(12), 1159.
- Venhuizen, N. J., Hendriks, P., Crocker, M. W., & Brouwer, H. (2022). Distributional formal semantics. *Information and Computation*, 287, 104925. <https://doi.org/10.1016/j.ic.2022.104925>
- Zwaan, R. A. (2004). The immersed experiencer: Toward an embodied theory of language comprehension. In B. H. Ross (Ed.), *The psychology of learning and motivation* (pp. 35–62, Vol. 44). Academic Press.

Zwaan, R. A., & Radvansky, G. A. (1998). Situation models in language comprehension and memory. *Psychological Bulletin*, 123(2), 162–185. <https://doi.org/10.1037/0033-2909.123.2.162>