# What Do Hebbian Learners Learn?
# Technical Appendix

**Anonymous submission**

## 1 Introduction

In this Supplemental Material file, we give much more detailed proofs for the claims we make in our paper. These proofs were developed in tandem with the Lean 4 interactive theorem prover (**?**).

In our paper submission, we accidentally provided the following web link to our Lean 4 code.

https://github.com/ais-climber/AAAI2024

We have since anonymized this webpage to the best of our ability, and notified the program chairs about this situation. This same code is also available as a Zip file, included with the supplemental materials.

We chose to develop these proofs with the help of a proof assistant mainly because the central results — Lemma **??** and Theorem **??** (our reduction theorem) — were difficult to get right by hand. So we prioritized formally verifying these main results first. Our vision was to have all of the supporting lemmas verified as well, but time ran out.

In light of this situation, we provide detailed English proofs of all the claims from our paper here. These proofs are self-contained, and may be checked in the same manner as if we did *not* formally verify any of it. If a claim has been formally verified in Lean, we mark it with V ("verified"); otherwise, we mark it with N ("not verified").

We give instructions for installing Lean and running our code at the end of this appendix (as well as in a README.md file included in the code). Additionally, we also comment on some of our Reproducibility Checklist items at the end of this appendix.

**Note.** N indicates that we have an English proof, but not a Lean 4 proof. In these situations we simply ran out of time transcribing them to Lean. For example, with Proposition **??**, Proposition **??**, Proposition **??**, and Lemma **??**, this was because of technical issues we had with reasoning about inequalities in Lean 4. We plan on fully verifying these claims in Lean by the AAAI-24 author feedback window. But for now, we recommend that these claims be given special attention (especially Proposition **??** and Lemma **??**, which are non-trivial).

## 2 Detailed Proofs for Section 2

**Proposition 1.** For all $m, n \in N$, N

$$m \in \text{preds}(n) \text{ iff } \mathsf{layer}(m) < \mathsf{layer}(n)$$

*Proof.*

(→) Suppose $m \in \text{preds}(n)$ and let $\mathsf{layer}(m) = k$. By definition, there is a path from some $x$ to $m$, where $\mathsf{layer}(x) = 0$. But since $m \in \text{preds}(n)$, we can extend this path to be from $x$ to $n$. And so $\mathsf{layer}(n) \geq k + 1$, i.e. $\mathsf{layer}(m) < \mathsf{layer}(n)$.

(←) Suppose $\mathsf{layer}(m) < \mathsf{layer}(n)$, but for contradiction $m$ is not a predecessor of $n$. Since $\mathcal{N}$ is fully connected, we have two possible cases (the third is eliminated, since $m \notin \text{preds}(n)$).

**Case 1.** $n \in \text{preds}(m)$. By the first part of Proposition **??**, $\mathsf{layer}(n) < \mathsf{layer}(m)$, which contradicts our assumption.

**Case 2.** $m$ and $n$ share exactly the same predecessors and successors. But this means they have exactly the same paths from any node, including nodes $x$ with $\mathsf{layer}(x) = 0$. By definition, $\mathsf{layer}(m) = \mathsf{layer}(n)$, which again contradicts our assumption.

□

**Proposition 2.** For all $S, S_1, \ldots, S_k \in \mathsf{State}$,

**Inclusion.** $S \subseteq \mathsf{Prop}(S)$ V

**Idempotence.** $\mathsf{Prop}(\mathsf{Prop}(S)) = \mathsf{Prop}(S)$ V

**Cumulative.** If $S_1 \subseteq S_2 \subseteq \mathsf{Prop}(S_1)$, then $\mathsf{Prop}(S_1) = \mathsf{Prop}(S_2)$ V

**Loop.** If $S_1 \subseteq \mathsf{Prop}(S_0), \ldots, S_k \subseteq \mathsf{Prop}(S_{k-1})$, and $S_0 \subseteq \mathsf{Prop}(S_k)$, then N

$$\mathsf{Prop}(S_i) = \mathsf{Prop}(S_j)$$

for all $i, j \in \{0, \ldots, k\}$

*Proof.* These properties come from (**?**). But as we mentioned in the paper, Leitgeb actually proves that the properties hold for a closure operator $Cl$ over *inhibition nets*, neural networks with (unweighted) excitatory and inhibitory connections. But at the end of (**?**) he shows that inhibition nets and (weighted) binary, feed-forward nets are equivalent. And so our $\mathsf{Prop}$ is identical to his $Cl$.

For completeness, and as a sanity check, we prove that these properties hold using our weighted-sum definition of $\mathsf{Prop}$. (We checked the first three in Lean 4 to check that our definitions were correct. But we weren't so worried about verifying (Loop), since it's somewhat awkward to express in Lean.)

**Inclusion.** Let $n \in S$. By induction on $\mathsf{layer}(n)$ in $\mathcal{N}$:

**Base Step.** $\mathsf{layer}(n) = 0$, and so $n \in S = \mathsf{Prop}(S)$.

**Inductive Step.** $\mathsf{layer}(n) \geq 0$. By definition of $\mathsf{Prop}$, since $n \in S$, $n \in \mathsf{Prop}(S)$.

**Idempotence.** For all $n \in N$, we show

$$n \in \mathsf{Prop}(\mathsf{Prop}(S)) \text{ iff } n \in \mathsf{Prop}(S)$$

by induction on $\mathsf{layer}(n)$.

**Base Step.** At layer 0, the statement simplifies to $S = S$, which is true.

**Inductive Step.** Let $\mathsf{layer}(n) \geq 0$.

(←) This direction is just Inclusion.

(→) Suppose $n \in \mathsf{Prop}(\mathsf{Prop}(S))$. We have two cases:

**Case 1.** $n \in \mathsf{Prop}(S)$. And so we have our goal.

**Case 2.** $n$ is activated by its predecessors $m_i \in \mathsf{Prop}(\mathsf{Prop}(S))$, i.e.

$$A(\sum_{i=1}^{\deg(n)} W(m_i, n) \cdot \chi_{\mathsf{Prop}(\mathsf{Prop}(S))}(m_i)) = 1$$

Our inductive hypothesis says that for all predecessors $m_i$, $m_i \in \mathsf{Prop}(\mathsf{Prop}(S)) \leftrightarrow m_i \in \mathsf{Prop}(S)$. So we substitute that in the inner expression:

$$A(\sum_{i=1}^{\deg(n)} W(m_i, n) \cdot \chi_{\mathsf{Prop}(S)}(m_i)) = 1$$

That is, $n \in \mathsf{Prop}(S)$.

**Cumulative.** Suppose $S_1 \subseteq S_2 \subseteq \mathsf{Prop}(S_1)$. For all $n \in N$, we show

$$n \in \mathsf{Prop}(S_1) \text{ iff } n \in \mathsf{Prop}(S_2)$$

by induction on $\mathsf{layer}(n)$.

**Base Step.** At layer 0, the statement reduces to $S_1 = S_2$. But this is true, since (at layer 0), our hypothesis reduces to $S_1 \subseteq S_2 \subseteq S_1$, which is only possible if $S_1 = S_2$.

**Inductive Step.** Let $\mathsf{layer}(n) \geq 0$.

($\rightarrow$) Suppose $n \in \mathsf{Prop}(S_1)$. We have two cases:

**Case 1.** $n \in S_1$. By our hypothesis, $n \in S_2$, and so by Inclusion $n \in \mathsf{Prop}(S_2)$.

**Case 2.** $n$ is activated by its predecessors $m_i \in \mathsf{Prop}(S_1)$, i.e.

$$A(\sum_{i=1}^{\deg(n)} W(m_i, n) \cdot \chi_{\mathsf{Prop}(S_1)}(m_i)) = 1$$

Our inductive hypothesis says that for all predecessors $m_i$, $m_i \in \mathsf{Prop}(S_1) \leftrightarrow m_i \in \mathsf{Prop}(S_2)$. So we substitute that in the inner expression:

$$A(\sum_{i=1}^{\deg(n)} W(m_i, n) \cdot \chi_{\mathsf{Prop}(S_2)}(m_i)) = 1$$

That is, $n \in \mathsf{Prop}(S_2)$.

($\leftarrow$) Suppose $n \in \mathsf{Prop}(S_2)$. We have two cases:

**Case 1.** $n \in S_2$. By our hypothesis, $n \in \mathsf{Prop}(S_1)$.

**Case 2.** $n$ is activated by its predecessors $m_i \in \mathsf{Prop}(S_2)$. Following exactly the same argument as the forward direction (swapping $S_1$ and $S_2$), we get $n \in \mathsf{Prop}(S_1)$.

**Loop.** Let $k \geq 0$ and suppose the hypothesis. Our goal is to show that for each $i$, $\mathsf{Prop}(S_i) \subseteq \mathsf{Prop}(S_{i-1})$, and additionally $\mathsf{Prop}(S_0) \subseteq \mathsf{Prop}(S_k)$. This will show that all $\mathsf{Prop}(S_i)$ contain each other, and so are equal. Let $i \in \{0, \dots, k\}$ (if $i = 0$ then $i - 1$ refers to $n$), and let $n \in \mathsf{Prop}(S_i)$. We proceed by induction on $\mathsf{layer}(n)$.

**Base Step.** At layer 0, $\mathsf{Prop}(S_i) = S_i$. And so $n \in S_i$. But $S_i \subseteq \mathsf{Prop}(S_{i-1})$ by our hypothesis, and so $n \in \mathsf{Prop}(S_{i-1})$.

**Inductive Step.** Let $\mathsf{layer}(n) \geq 0$. Since $n \in \mathsf{Prop}(S_i)$, we have two cases:

**Case 1.** $n \in S_i$. By our hypothesis, $n \in \mathsf{Prop}(S_{i-1})$.

**Case 2.** $n$ is activated by its predecessors $m_i \in \mathsf{Prop}(S_i)$, i.e.

$$A(\sum_{h=1}^{\deg(n)} W(m_h, n) \cdot \chi_{\mathsf{Prop}(S_i)}(m_h)) = 1$$

Our inductive hypothesis says that for all predecessors $m_h$ and all $i, j$, $m_h \in \mathsf{Prop}(S_i) \leftrightarrow m_h \in \mathsf{Prop}(S_j)$. In particular, this is true for $S_i$ and $S_{i-1}$. So we substitute that in the inner expression:

$$A(\sum_{h=1}^{\deg(n)} W(m_h, n) \cdot \chi_{\mathsf{Prop}(S_{i-1})}(m_h)) = 1$$

That is, $n \in \mathsf{Prop}(S_{i-1})$.

□

**Proposition 3.** For all $A, B, C \in \mathsf{State}$,

**Layer** $0$. If $\mathsf{layer}(n) = 0$, V
$n \in \mathsf{Reach}(A, B)$ implies $n \in B$

**Empty.** If $A \cap B = \emptyset$ then $\mathsf{Reach}(A, B) = \emptyset$ V

**Subsumption.** $\mathsf{Reach}(A, B) \subseteq A$ V

**Inclusion.** $A \cap B \subseteq \mathsf{Reach}(A, B)$ V

**Idempotent.** V
$\mathsf{Reach}(A, \mathsf{Reach}(A, B)) = \mathsf{Reach}(A, B)$

**Monotonic.** V
If $B \subseteq C$ then $\mathsf{Reach}(A, B) \subseteq \mathsf{Reach}(A, C)$

**Closed under union.** V
$\mathsf{Reach}(A, B \cup C) = \mathsf{Reach}(A, B) \cup \mathsf{Reach}(A, C)$

*Proof.* We prove each in turn:

**Layer** $0$. Suppose $\mathsf{layer}(n) = 0$ and $n \in \mathsf{Reach}(A, B)$. So there is a path from some $x \in B$ to $n$ running entirely through $A$. But by definition $\mathsf{layer}(n) = 0$ means that $n$ has no predecessors! So this path must be from $n \in B$ to itself. And so $n \in B$.

**Empty.** Suppose contrapositively that some $n \in \mathsf{Reach}(A, B)$. So there is a path from some $m \in B$ to $n$ running entirely through $A$. In particular, $m \in A$. But this means that $m \in A \cap B$.

**Subsumption.** If $n \in \mathsf{Reach}(A, B)$, then there is a path from some $m \in B$ to $n$ running entirely through $A$. In particular, $n \in A$.

**Inclusion.** If $n \in A \cap B$, then there is a path from $n \in B$ to itself running entirely through $A$. And so $n \in \mathsf{Reach}(A, B)$.

**Idempotent.**

($\rightarrow$) Suppose $n \in \mathsf{Reach}(A, \mathsf{Reach}(A, B))$. Then there is a path from some $y \in \mathsf{Reach}(A, B)$ to $n$ running entirely through $A$. But again, there is a path from some $x \in B$ to $y$ running entirely through $A$. We can combine these paths to get a new path from $x$ to $n$. And so $n \in \mathsf{Reach}(A, B)$.

($\leftarrow$) Suppose $n \in \mathsf{Reach}(A, B)$. Then there is a path from some $x \in B$ to $n$ running entirely through $A$. But $x \in \mathsf{Reach}(A, B)$ (via the path to itself), and so we can use this same path to show that $n \in \mathsf{Reach}(A, \mathsf{Reach}(A, B))$.

**Monotonic.** Suppose $B \subseteq C$. If $n \in \mathsf{Reach}(A, B)$, then there is a path from some $m \in B$ to $n$ running entirely through $A$. But since $B \subseteq C$, we have $m \in C$. And so $n \in \mathsf{Reach}(A, C)$.

**Closed under union.**

($\supseteq$) This direction follows from monotonicity. Without loss of generality, say $n \in \mathsf{Reach}(A, B)$. Well, $B \subseteq B \cup C$, and since $\mathsf{Reach}$ is monotonic we have $n \in \mathsf{Reach}(A, B \cup C)$.

($\subseteq$) Now suppose $n \in \mathsf{Reach}(A, B \cup C)$. So there is a path from some $m \in B \cup C$ to $n$ running entirely through $A$. So either $n \in B$, and so $n \in \mathsf{Reach}(A, B)$; or $n \in C$, and so $n \in \mathsf{Reach}(A, C)$.

□

## 3 Detailed Proofs for Section 3

**Proposition 4.** For all $S, A, B \in \mathsf{State}$, V

$$\mathsf{Reach}_{\mathsf{Hebb}(\mathcal{N}, S)}(A, B) = \mathsf{Reach}_{\mathcal{N}}(A, B)$$

*Proof.* A single step of Hebbian update $\mathsf{Hebb}$ doesn't change the edge relation $E$ of the graph (it only changes the weights of the graph). So if $n \in N$, any path from $m \in B$ to $n$ running entirely through $A$ in $\mathsf{Hebb}(\mathcal{N}, S)$ is the same path in $\mathcal{N}$. □

**Proposition 5.** Let $S \in \mathsf{State}$, $m, n \in N$. We have:

- $W_{\mathcal{N}}(m,n) \leq W_{\mathsf{Hebb}(\mathcal{N},S)}(m,n)$ N
- If either $m \notin \mathsf{Prop}(S)$ or $n \notin \mathsf{Prop}(S)$, then V

$$W_{\mathsf{Hebb}(\mathcal{N},S)}(m,n) = W_{\mathcal{N}}(m,n)$$

*Proof.* For the first part, observe:

$$\begin{aligned} W_{\mathcal{N}}(m,n) &\leq W_{\mathcal{N}}(m,n) + \eta \qquad (\text{since } \eta \geq 0) \\ &\leq W_{\mathcal{N}}(m,n) + \eta \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m) \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n) \\ &\qquad (\text{since } \forall S, n, \chi_S(n) \geq 0) \\ &= W_{\mathsf{Hebb}(\mathcal{N},S)}(m,n) \end{aligned}$$

As for the second part, if either $m \notin \mathsf{Prop}(S)$ or $n \notin \mathsf{Prop}(S)$, then by definition of $\mathsf{Hebb}$,

$$\begin{aligned} &W_{\mathsf{Hebb}(\mathcal{N},S)}(m,n) \\ &\quad = W_{\mathcal{N}}(m,n) + \eta \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m) \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n) \\ &\quad = W_{\mathcal{N}}(m,n) + \eta \cdot 0 \\ &\quad = W_{\mathcal{N}}(m,n) + 0 \\ &\quad = W_{\mathcal{N}}(m,n) \end{aligned}$$

□

**Proposition 6.** For all $S \in \mathsf{State}$, $m, n \in N$, we have N

$$\mathsf{mnws} \leq W(m,n) \cdot \chi_S(m)$$

*Proof.* Let $m, n$ be any nodes in $N$. We have:

$$\begin{aligned} &\mathsf{mnws} \\ &\quad \leq \mathsf{nws}(n) \\ &\quad = \textstyle\sum_{i=1}^{\deg(n)} \begin{cases} W(m_i,n) & \text{if } W(m_i,n) < 0 \\ 0 & \text{otherwise} \end{cases} \\ &\qquad (\text{by definition}) \\ &\quad \leq \textstyle\sum_{i=1}^{\deg(n)} \begin{cases} W(m_i,n) \cdot \chi_S(m_i) & \text{if } W(m_i,n) < 0 \\ 0 & \text{otherwise} \end{cases} \\ &\qquad (\text{since each } W(m_i,n) < 0 \text{ and } \chi_S(m_i) \in \{0,1\}) \\ &\quad \leq W(m,n) \cdot \chi_S(m) \\ &\qquad (\text{the sum of negative terms is } \leq \text{ any} \\ &\qquad\quad \text{particular term}) \end{aligned}$$

□

**Proposition 7.** For all $S \in \mathsf{State}$, N

$$\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S)}(S) = \mathsf{Prop}_{\mathcal{N}}(S)$$

*Proof.* For all $n \in N$, we show

$$n \in \mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S)}(S) \text{ iff } n \in \mathsf{Prop}_{\mathcal{N}}(S)$$

by induction on $\mathsf{layer}(n)$.

**Base Step.** At layer 0, the statement simplifies to $A = A$, which is true.

**Inductive Step.** Let $\mathsf{layer}(n) \geq 0$.

($\leftarrow$) Suppose $n \in \mathsf{Prop}_{\mathcal{N}}(S)$. We have two cases:

**Case 1.** $n \in S$. By inclusion, $n \in \mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S)}(S)$.

**Case 2.** $n$ is activated by its predecessors $m_i$, i.e.

$$A(\sum_{i=1}^{\deg(n)} W_{\mathcal{N}}(m_i,n) \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m_i)) = 1$$

by inductive hypothesis, we can substitute in the inner expression:

$$A(\sum_{i=1}^{\deg(n)} W_{\mathcal{N}}(m_i,n) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S)}(S)}(m_i)) = 1$$

By the first part of Proposition **??**, $W_{\mathcal{N}}(m_i,n) \leq W_{\mathsf{Hebb}(\mathcal{N},S)}(m_i,n)$. So the inner sum using the former is $\leq$ the inner sum using the latter. Since $A$ is nondecreasing, we have

$$A(\sum_{i=1}^{\deg(n)} W_{\mathsf{Hebb}(\mathcal{N},S)}(m_i,n) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S)}(S)}(m_i)) = 1$$

i.e. $n \in \mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S)}(S)$.

($\rightarrow$) Suppose $n \in \mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S)}(S)$. Again, we have two cases:

**Case 1.** $n \in S$. By inclusion, $n \in \mathsf{Prop}_{\mathcal{N}}(S)$.

**Case 2.** $n$ is activated by its predecessors $m_i$, i.e.

$$A(\sum_{i=1}^{\deg(n)} W_{\mathsf{Hebb}(\mathcal{N},S)}(m_i,n) \cdot \chi_{\mathsf{Prop}_{\mathsf{Hebb}(\mathcal{N},S)}(S)}(m_i)) = 1$$

By inductive hypothesis, we can substitute in the inner expression:

$$A(\sum_{i=1}^{\deg(n)} W_{\mathsf{Hebb}(\mathcal{N},S)}(m_i,n) \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m_i)) = 1$$

Now suppose for contradiction that $n \notin \mathsf{Prop}_{\mathcal{N}}(S)$. By the second part of Proposition **??**, $W_{\mathsf{Hebb}(\mathcal{N},S)}(m_i,n) = W_{\mathcal{N}}(m_i,n)$, and so we have

$$A(\sum_{i=1}^{\deg(n)} W_{\mathcal{N}}(m_i,n) \cdot \chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m_i)) = 1$$

i.e. $n \in \mathsf{Prop}_{\mathcal{N}}(S)$, which contradicts $n \notin \mathsf{Prop}_{\mathcal{N}}(S)$.

□

**Proposition 8.** For all $S, A, B \in \mathsf{State}$, N

$$\mathsf{Reach}_{\mathsf{Hebb}^*(\mathcal{N},S)}(A,B) = \mathsf{Reach}_{\mathcal{N}}(A,B)$$

*Proof.* As with $\mathsf{Hebb}$, iterated Hebbian update $\mathsf{Hebb}^*$ doesn't change the edge relation $E$ of the graph (it only changes the weights of the graph). So if $n \in N$, any path from $m \in B$ to $n$ running entirely through $A$ in $\mathsf{Hebb}^*(\mathcal{N}, S)$ is the same path in $\mathcal{N}$. □

**Proposition 9.** Let $m, n \in N$. We have:

- $W_{\mathcal{N}}(m,n) \leq W_{\mathsf{Hebb}^*(\mathcal{N},S)}(m,n)$ N
- If either $m \notin \mathsf{Prop}(S)$ or $n \notin \mathsf{Prop}(S)$, then N

$$W_{\mathsf{Hebb}^*(\mathcal{N},S)}(m,n) = W_{\mathcal{N}}(m,n)$$

*Proof.* First, notice that if $\mathsf{iter} = 1$,

$$\begin{aligned} &W_{\mathsf{Hebb}^*(\mathcal{N},S)}(m,n) \\ &\quad = W_{\mathcal{N}}(m,n) + \mathsf{iter}\cdot\eta\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m)\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n) \\ &\quad = W_{\mathcal{N}}(m,n) + 1\cdot\eta\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m)\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n) \\ &\quad = W_{\mathcal{N}}(m,n) + \eta\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m)\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n) \\ &\quad = W_{\mathsf{Hebb}(\mathcal{N},S)}(m,n) \end{aligned}$$

i.e. $W_{\mathsf{Hebb}^*(\mathcal{N},S)}(m,n)$ reduces to $W_{\mathsf{Hebb}(\mathcal{N},S)}(m,n)$

Now for the first claim above. We proceed by induction on $\mathsf{iter}$.

**Base Step.** $\mathsf{iter} = 1$, and so the claim reduces to the first part of Proposition **??**.

**Inductive Step.** $\mathsf{iter} \geq 1$. We take as our Inductive Hypothesis that for all $k < \mathsf{iter}$:

$$W_{\mathcal{N}}(m,n) \leq W_{\mathcal{N}}(m,n) + k\cdot\eta\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m)\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n)$$

Since the term

$$k\cdot\eta\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m)\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n) \geq 0$$

and $1 \leq k < \mathsf{iter}$, we have

$$W_{\mathcal{N}}(m,n) \leq W_{\mathcal{N}}(m,n) + \mathsf{iter}\cdot\eta\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m)\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n)$$

i.e.

$$W_{\mathcal{N}}(m,n) \leq W_{\mathsf{Hebb}^*(\mathcal{N},S)}(m,n)$$

As for the second part, if either $m \notin \mathsf{Prop}(S)$ or $n \notin \mathsf{Prop}(S)$, then just as with $\mathsf{Hebb}$, we have for $\mathsf{Hebb}^*$

$$\begin{aligned} &W_{\mathsf{Hebb}^*(\mathcal{N},S)}(m,n) \\ &\quad = W_{\mathcal{N}}(m,n) + \mathsf{iter}\cdot\eta\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(m)\cdot\chi_{\mathsf{Prop}_{\mathcal{N}}(S)}(n) \\ &\quad = W_{\mathcal{N}}(m,n) + \mathsf{iter}\cdot\eta\cdot 0 \\ &\quad = W_{\mathcal{N}}(m,n) + 0 \\ &\quad = W_{\mathcal{N}}(m,n) \end{aligned}$$

□

**Lemma 1.** Let $A, B \in \mathsf{State}$ and $m, n \in N$. Suppose $m \in \mathsf{preds}(n)$, $m, n \in \mathsf{Prop}(A)$, $m \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$. Then N

$$A(\sum_{i=1}^{\deg(n)} W_{\mathsf{Hebb}^*(\mathcal{N},A)}(m_i,n)\cdot\chi_{\mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)}(m_i)) = 1$$

*Proof.* Recall that $A$ has a threshold, i.e., $\exists t \in \mathbb{Q}$ with $A(t) = 1$. Since $A$ is nondecreasing, it's enough for us to show

$$t \leq \sum_{i=1}^{\deg(n)} W_{\mathsf{Hebb}^*(\mathcal{N},A)}(m_i,n)\cdot\chi_{\mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)}(m_i)$$

Without loss of generality, say $m \in \mathsf{preds}(n)$ is the $(\deg(n)-1)^{\text{th}}$ predecessor of $n$ (if it is not, just re-label the predecessors of $n$ so that it is). Then we have

$$\begin{aligned} &\textstyle\sum_{i=1}^{\deg(n)} W_{\mathsf{Hebb}^*(\mathcal{N},A)}(m_i,n)\cdot\chi_{\mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)}(m_i) \\ &\quad = \textstyle\sum_{i=1}^{\deg(n)-1} W_{\mathsf{Hebb}^*(\mathcal{N},A)}(m_i,n)\cdot\chi_{\mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)}(m_i) \\ &\qquad + W_{\mathsf{Hebb}^*(\mathcal{N},A)}(m,n)\cdot\chi_{\mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)}(m) \\[1ex] &\quad \geq (|N|-1)\cdot\mathsf{mnws} \\ &\qquad + W_{\mathsf{Hebb}^*(\mathcal{N},A)}(m,n)\cdot\chi_{\mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)}(m) \\ &\qquad \text{(by Proposition ??, since we are adding } |N|-1 \text{ terms)} \\[1ex] &\quad = (|N|-1)\cdot\mathsf{mnws} + W_{\mathsf{Hebb}^*(\mathcal{N},A)}(m,n)\cdot 1 \\ &\qquad \text{(since } m \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)\text{)} \\[1ex] &\quad = (|N|-1)\cdot\mathsf{mnws} \\ &\qquad + W_{\mathcal{N}}(m,n) + \mathsf{iter}\cdot\eta\cdot\chi_{\mathsf{Prop}(A)}(m)\cdot\chi_{\mathsf{Prop}(A)}(n) \\ &\qquad \text{(by definition of } \mathsf{Hebb}^*\text{)} \\[1ex] &\quad = (|N|-1)\cdot\mathsf{mnws} + W_{\mathcal{N}}(m,n) + \mathsf{iter}\cdot\eta \\ &\qquad \text{(since } m, n \in \mathsf{Prop}(A)\text{)} \\[1ex] &\quad \geq (|N|-1)\cdot\mathsf{mnws} + \mathsf{mnws} + \mathsf{iter}\cdot\eta \\ &\qquad \text{(the sum of negative weights is } \leq \text{ any} \\ &\qquad\quad \text{particular weight)} \\[1ex] &\quad \geq |N|\cdot\mathsf{mnws} + \mathsf{iter}\cdot\eta \\ &\qquad \text{(grouping like terms)} \end{aligned}$$

So we need to show:

$$t \leq |N|\cdot\mathsf{mnws} + \mathsf{iter}\cdot\eta$$

Rearranging this to solve for iter, it suffices to show:

$$\frac{t - |N|\cdot\mathsf{mnws}}{\eta} \leq \mathsf{iter}$$

But we defined iter to be exactly the integer ceiling of this expression on the left (and 1 if the expression on the left is negative)! □

## 4 Detailed Proofs for Section 4

**Lemma 2.** Let $A, B \in \mathsf{State}$. $\mathsf{Hebb}^*$ satisfies the following algebraic properties.

1. $\mathsf{Prop}(A) \cap \mathsf{Prop}(B) \subseteq \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$ V
2. $\mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)) \subseteq \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$ V
3. $\mathsf{Prop}(A) \cap \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B) \subseteq \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$ V

*Proof.* We prove each in turn. We plan on using (2) and (3) in the proof of our reduction, but we need (1) in order to prove (2). (1) and (3) crucially depend on our assumption that the net $\mathcal{N}$ is fully connected.

1. Suppose $n \in \mathsf{Prop}(A) \cap \mathsf{Prop}(B)$. We proceed by induction on $\mathsf{layer}(n)$.

   **Base Step.** At layer 0, the statement simplifies to $A \cap B \subseteq B$, which is true.

   **Inductive Step.** Let $\mathsf{layer}(n) \geq 0$. From $n \in \mathsf{Prop}(B)$ we have two cases:

   **Case 1.** $n \in B$. By Inclusion, we have

   $$n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$$

   **Case 2.** $n$ is activated by its predecessors $m_i \in \mathsf{Prop}(B)$ in $\mathcal{N}$. Note that since $\mathsf{Prop}(A) \cap \mathsf{Prop}(B) \neq \emptyset$, by well-ordering there is some $m \in \mathsf{Prop}(A) \cap \mathsf{Prop}(B)$ with the smallest layer. In particular, $\mathsf{layer}(m) \leq \mathsf{layer}(n)$. From here, we have two more cases:

**Case 2.1.** $\mathsf{layer}(m) < \mathsf{layer}(n)$. Since our net is fully connected, this means that $m \in \text{preds}(n)$. Applying our inductive hypothesis to $m$, we have

$$m \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$$

And since $m, n \in \mathsf{Prop}(A)$, $m \in \text{preds}(n)$, we now have exactly the right conditions for Lemma **??**! From this it follows that

$$n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$$

**Case 2.2.** $\mathsf{layer}(m) = \mathsf{layer}(n)$. Since $m$ is the smallest such node in $\mathsf{Prop}(A) \cap \mathsf{Prop}(B)$, and it is not a predecessor of $n$, none of $n$'s predecessors are in $\mathsf{Prop}(A) \cap \mathsf{Prop}(B)$. From this and Proposition **??** we can inductively argue that the weights of the *active* predecessors in $\mathsf{Hebb}^*(\mathcal{N}, A)$ are the same as their weights in $\mathcal{N}$. And so we get

$$n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$$

2. In the process of transcribing the proofs from the Lean code, we found a simpler way to prove this. So this proof is slightly different from the one presented in the code. Suppose $n \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$. By definition, there is a path from some $x \in \mathsf{Prop}(B)$ to $n$ running entirely through $\mathsf{Prop}(A)$. By induction on the length of this path:

**Base Step.** The path is from $n$ to itself. But this means that $n \in \mathsf{Prop}(A) \cap \mathsf{Prop}(B)$. By Part (1) of this lemma, we have $n \in \mathsf{Prop}^*_{\mathsf{Hebb}}(\mathcal{N}, A)(B)$.

**Inductive Step.** Say this path is from $x$ to $m$, and $m \in \text{preds}(n)$. Since $m \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$, we can apply our inductive hypothesis on $m$ to get

$$m \in \mathsf{Prop}^*_{\mathsf{Hebb}}(\mathcal{N}, A)(B)$$

And since $m, n \in \mathsf{Prop}(A)$, $m \in \text{preds}(n)$, we now have exactly the right conditions for Lemma **??**! From this it follows that

$$n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$$

3. Suppose $n \in \mathsf{Prop}(A) \cap \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$. We proceed by induction on $\mathsf{layer}(n)$.

**Base Step.** At layer 0, the hypothesis simplifies to $n \in A \cap B$. By $\mathsf{Prop}$-Inclusion, $n \in \mathsf{Prop}(A)$ and $n \in \mathsf{Prop}(B)$. By $\mathsf{Reach}$-Inclusion, $n \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$.

**Inductive Step.** Let $\mathsf{layer}(n) \geq 0$. From $n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$ we have two cases:

**Case 1.** $n \in B$. By $\mathsf{Prop}$-Inclusion, $n \in \mathsf{Prop}(B)$. And so $n \in \mathsf{Prop}(A) \cap \mathsf{Prop}(B)$, from which we conclude $n \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$ by $\mathsf{Reach}$-Inclusion.

**Case 2.** $n$ is activated by its predecessors $m_i \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$ in $\mathsf{Hebb}^*(\mathcal{N}, A)$. Note that since $\mathsf{Prop}(A) \cap \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B) \neq \emptyset$, by well-ordering there is some $m \in \mathsf{Prop}(A) \cap \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$ with the smallest layer. In particular, $\mathsf{layer}(m) \leq \mathsf{layer}(n)$. From here, we have two more cases:

**Case 2.1.** $\mathsf{layer}(m) < \mathsf{layer}(n)$. Since our net is fully connected, this means that $m \in \text{preds}(n)$. Applying our inductive hypothesis to $m$, we have

$$m \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$$

By definition of $\mathsf{Reach}$, we have a path from some $x \in \mathsf{Prop}(B)$ to $m$ running entirely through $\mathsf{Prop}(A)$. But since $m, n \in \mathsf{Prop}(A)$, $m \in \text{preds}(n)$, we can extend this path to $n$. So

$$n \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$$

**Case 2.2.** $\mathsf{layer}(m) = \mathsf{layer}(n)$. Since $m$ is the smallest such node in $\mathsf{Prop}(A) \cap \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$, and it is not a predecessor of $n$, none of $n$'s predecessors are in $\mathsf{Prop}(A) \cap \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$. From this we apply Proposition **??** inductively as before to argue that the weights of the *active* predecessors in $\mathsf{Hebb}^*(\mathcal{N}, A)$ are the same as their weights in $\mathcal{N}$. And so we get $n \in \mathsf{Prop}(B)$. But then we have $n \in \mathsf{Prop}(A) \cap \mathsf{Prop}(B)$, so by $\mathsf{Reach}$-Inclusion we have $n \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$.

□

**Theorem 3 (Reduction).** For all $A, B \in \mathsf{State}$, V

$$\mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B) = \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$$

*Proof.* For all $n \in N$, we show

$$n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B) \quad \text{iff } n \in \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$$

by induction on $\mathsf{layer}(n)$.

**Base Step.** At layer 0, the outer $\mathsf{Prop}$'s simplify, and we need to show

$$B = B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$$

The $\subseteq$ direction is easy. For the $\supseteq$ direction, suppose $n \in B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$. If $n \in B$, we're done. On the other hand, if $n \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$ then $n \in B$ by the Layer 0 property for $\mathsf{Reach}$.

**Inductive Step.** Let $\mathsf{layer}(n) \geq 0$.

($\rightarrow$) Suppose $n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$. We have two cases:

**Case 1.** $n \in B$. So $n \in B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$. By Inclusion for $\mathsf{Prop}$ we have our goal.

**Case 2.** $n$ is activated by its predecessors $m_i \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$ in $\mathsf{Hebb}^*$. From here we split into two more cases:

**Case 2.1.** $n \in \mathsf{Prop}(A)$ and $\exists m \in \text{preds}(n)$ such that

$$m \in \mathsf{Prop}(A) \cap \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$$

by our inductive hypothesis on $m$, we have

$$m \in \mathsf{Prop}(A) \cap \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$$

So by Part 3 of Lemma **??**,

$$m \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$$

By definition of $\mathsf{Reach}$, there is a path from some $x \in \mathsf{Prop}(B)$ to $m$ running entirely through $\mathsf{Prop}(A)$. But since $m, n \in \mathsf{Prop}(A)$, we can extend this path to $n$. And so

$$n \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$$

By Inclusion, we conclude that

$$n \in \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$$

**Case 2.2.** Either $n \notin \mathsf{Prop}(A)$ or $\forall m \in \text{preds}(n)$, either $m \notin \mathsf{Prop}(A)$ or $m$ is not active ($m \notin \mathsf{Prop}(A) \cap \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$). In either case, by Proposition **??**, the weights of the two nets are the same. From here, we just substitute our inductive hypothesis into the weighted sum, and we have

$$n \in \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$$

($\leftarrow$) Suppose $n \in \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$. We have two cases:

**Case 1.** $n \in B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$. If $n \in B$, we're done (by Inclusion). On the other hand, if $n \in \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B))$, then by Part 2 of Lemma **??**, $n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$.

**Case 2.** $n$ is activated by its predecessors $m_i \in \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$ in $\mathcal{N}$. This case is easy — by Proposition **??** the weights in the new net are $\geq$ the weights for the old net, so we have the right weighted sum for $n$ being activated by its predecessors $m_i \in \mathsf{Prop}(B \cup \mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)))$ in $\mathsf{Hebb}^*(\mathcal{N}, A)$ (note that only the net has changed). From here, we just substitute our inductive hypothesis into the weighted sum, and we have

$$n \in \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B)$$

□

**Corollary 1.** If $\mathsf{Prop}(A) \cap \mathsf{Prop}(B) = \emptyset$ then V

$$\mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B) = \mathsf{Prop}(B)$$

*Proof.* Note that by the **Empty** property of $\mathsf{Reach}$, $\mathsf{Prop}(A) \cap \mathsf{Prop}(B) = \emptyset$ implies $\mathsf{Reach}(\mathsf{Prop}(A), \mathsf{Prop}(B)) = \emptyset$. And so by Theorem **??**,

$$\begin{array}{rcl} \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},A)}(B) & = & \mathsf{Prop}(B \cup \emptyset) \\ & = & \mathsf{Prop}(B) \end{array}$$

□

**Theorem 4 (Reduction Axioms).** The following axioms for $[\varphi]$ are sound. N

$$\begin{array}{lcll} \langle\varphi\rangle p & \leftrightarrow & p & \text{for propositions } p \\ \langle\varphi\rangle\neg\psi & \leftrightarrow & \neg\langle\varphi\rangle\psi & \\ \langle\varphi\rangle(\psi \wedge \rho) & \leftrightarrow & \langle\varphi\rangle\psi \wedge \langle\varphi\rangle\rho & \\ \langle\varphi\rangle\langle\mathbf{K}\rangle(\psi, \rho) & \leftrightarrow & \langle\mathbf{K}\rangle(\langle\varphi\rangle\psi, \langle\varphi\rangle\rho) & \\ \langle\varphi\rangle\langle\mathbf{B}\rangle\psi & \leftrightarrow & \langle\mathbf{B}\rangle(\langle\varphi\rangle\psi \vee \langle\mathbf{K}\rangle(\langle\mathbf{B}\rangle\varphi, \langle\mathbf{B}\rangle\langle\varphi\rangle\psi)) & \end{array}$$

*Proof.* We prove each are sound in turn. Soundness for $\varphi$ just means that for all interpreted nets $\mathcal{N} \in \mathsf{Net}$, $\mathcal{N} \models \varphi$. In other words, $\llbracket\varphi\rrbracket_{\mathcal{N}} = N$. To prove that an equivalence $\varphi \leftrightarrow \psi$ is sound, it is enough to show

$$\llbracket\varphi\rrbracket_{\mathcal{N}} = \llbracket\psi\rrbracket_{\mathcal{N}}$$

The first three cases are routine. The $\langle\mathbf{K}\rangle$ case corresponds to Proposition **??** for $\mathsf{Reach}$. The $\langle\mathbf{B}\rangle$ case is the most crucial, and corresponds to our Reduction (Theorem **??**). In order to greatly simplify the proofs, we just write $\llbracket\varphi\rrbracket$ for $\llbracket\varphi\rrbracket_{\mathcal{N}}$ (i.e. $\mathcal{N}$ without update).

**$p$ case.** Recall that for propositions $p$, $\mathsf{Hebb}^*$ does not change the interpretation $\llbracket p\rrbracket$. So

$$\llbracket\langle\varphi\rangle p\rrbracket = \llbracket p\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)} = \llbracket p\rrbracket$$

**$\neg\psi$ case.** We have

$$\begin{array}{rcl} \llbracket\langle\varphi\rangle\neg\psi\rrbracket & = & \llbracket\neg\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)} \\ & = & (\llbracket\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)})^{\mathsf{C}} \\ & = & (\llbracket\langle\varphi\rangle\psi\rrbracket)^{\mathsf{C}} \\ & = & \llbracket\neg\langle\varphi\rangle\psi\rrbracket \end{array}$$

**$\psi \wedge \rho$ case.** We have

$$\begin{array}{rcl} \llbracket\langle\varphi\rangle(\psi \wedge \rho)\rrbracket & = & \llbracket\psi \wedge \rho\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)} \\ & = & \llbracket\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)} \cap \llbracket\rho\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)} \\ & = & \llbracket\langle\varphi\rangle\psi\rrbracket \cap \llbracket\langle\varphi\rangle\rho\rrbracket \\ & = & \llbracket\langle\varphi\rangle\psi \wedge \langle\varphi\rangle\rho\rrbracket \end{array}$$

**$\langle\mathbf{K}\rangle(\psi, \rho)$ case.** This case looks a bit hectic, but all we're doing is decomposing our semantics until we can apply Proposition **??**.

$$\begin{array}{l} \llbracket\langle\varphi\rangle\langle\mathbf{K}\rangle(\psi, \rho)\rrbracket \\ = \llbracket\langle\mathbf{K}\rangle(\psi, \rho)\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)} \\ = \mathsf{Reach}_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)}(\llbracket\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)}, \llbracket\rho\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)}) \\ = \mathsf{Reach}(\llbracket\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)}, \llbracket\rho\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)}) \\ = \mathsf{Reach}(\llbracket\langle\varphi\rangle\psi\rrbracket, \llbracket\langle\varphi\rangle\rho\rrbracket) \\ = \llbracket\langle\mathbf{K}\rangle(\langle\varphi\rangle\psi, \langle\varphi\rangle\rho)\rrbracket \end{array}$$

**$\langle\mathbf{B}\rangle\psi$ case.** This case also looks hectic, but again we just decompose our semantics and apply Theorem **??**.

$$\begin{array}{l} \llbracket\langle\varphi\rangle\langle\mathbf{B}\rangle\psi\rrbracket \\ \quad = \llbracket\langle\mathbf{B}\rangle\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)} \\ \quad = \mathsf{Prop}_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)}(\llbracket\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)}) \\ \quad = \mathsf{Prop}(\llbracket\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)} \cup \\ \quad \mathsf{Reach}(\mathsf{Prop}(\llbracket\varphi\rrbracket), \mathsf{Prop}(\llbracket\psi\rrbracket_{\mathsf{Hebb}^*(\mathcal{N},\llbracket\varphi\rrbracket)}))) \\ \quad = \mathsf{Prop}(\llbracket\langle\varphi\rangle\psi\rrbracket \cup \\ \quad \mathsf{Reach}(\mathsf{Prop}(\llbracket\varphi\rrbracket), \mathsf{Prop}(\llbracket\langle\varphi\rangle\psi\rrbracket))) \\ \quad = \llbracket\langle\mathbf{B}\rangle(\langle\varphi\rangle\psi \vee \langle\mathbf{K}\rangle(\langle\mathbf{B}\rangle\varphi, \langle\mathbf{B}\rangle\langle\varphi\rangle\psi))\rrbracket \end{array}$$

□

**Theorem 5 (Model Building).** Suppose that we have model building for our static language $\mathcal{L}$, i.e. for all consistent $\Gamma \subseteq \mathcal{L}$ there is a net $\mathcal{N} \in \mathsf{Net}$ such that $\mathcal{N} \models \Gamma$. Then we have model building for our dynamic language as well: for all $\Gamma^* \subseteq \mathcal{L}^*$, there is $\mathcal{N}$ such that $\mathcal{N} \models \Gamma^*$. N

*Proof Sketch.* Let $\Gamma^* \subseteq \mathcal{L}^*$. As outlined in the paper, our plan is to define rewrite rules based on our reduction axioms that "translate away" all of the dynamic formulas $\langle\varphi\rangle\psi$ in $\Gamma^*$, resulting in $\Gamma^{\text{tr}} \subseteq \mathcal{L}$. By our assumption, we have a net $\mathcal{N} \models \Gamma^{\text{tr}}$, and we show that this very same net $\mathcal{N} \models \Gamma^*$.

It's easy to see intuitively how this translation should go. For example, given the formula

$$\langle p\rangle(\langle p\rangle\langle\mathbf{B}\rangle q \wedge \langle\mathbf{K}\rangle(p, q)) \in \Gamma^*$$

we would recursively apply our reduction axioms, pushing $\langle p\rangle$ further into the expression until we can eliminate the propositional cases $\langle p\rangle q$.

We define the term-rewriting system that does the translation $tr(\varphi)$ for all $\varphi$ as follows.

- $tr(p) = p$
- $tr(\neg\varphi) = \neg tr(\varphi)$
- $tr(\varphi \wedge \psi) = tr(\varphi) \wedge tr(\psi)$
- $tr(\mathbf{K}(\varphi, \psi)) = \mathbf{K}(tr(\varphi), tr(\psi))$
- $tr(\langle\varphi\rangle p) = tr(p)$
- $tr(\langle\varphi\rangle\neg\psi) = tr(\neg\langle\varphi\rangle\psi)$
- $tr(\langle\varphi\rangle(\psi \wedge \rho)) = tr(\langle\varphi\rangle\psi \wedge \langle\varphi\rangle\rho)$
- $tr(\langle\varphi\rangle\langle\mathbf{K}\rangle(\psi, \rho)) = tr(\langle\mathbf{K}\rangle(\langle\varphi\rangle\psi, \langle\varphi\rangle\rho))$
- $tr(\langle\varphi\rangle\langle\mathbf{B}\rangle\psi) = tr(\langle\mathbf{B}\rangle(\langle\varphi\rangle\psi \vee \langle\mathbf{K}\rangle(\langle\mathbf{B}\rangle\varphi, \langle\mathbf{B}\rangle\langle\varphi\rangle\psi)))$
- $tr(\langle\varphi\rangle\langle\psi\rangle\rho) = tr(\langle\varphi\rangle(tr(\langle\psi\rangle\rho)))$

Formally, the term-rewriting system takes a formula $\varphi$ and recursively applies these equational rules to $\varphi$ (from left-to-right). We just need to check that

1. For all $\psi$, $tr(\psi)$ is update-operator-free
2. This term rewriting actually terminates

The work involved in showing termination is long and tedious. The usual approach is to define a measure on formulas $c(\varphi)$ that *decreases* with each application of our reduction axioms (from left-to-right). In particular, we need $c$ to satisfy

- If $\psi$ is a subexpression of $\varphi$, $c(\varphi) > c(\psi)$

- $c(\langle\varphi\rangle p) > c(p)$
- $c(\langle\varphi\rangle\neg\psi) > c(\neg\langle\varphi\rangle\psi)$
- $c(\langle\varphi\rangle(\psi \wedge \rho)) > c(\langle\varphi\rangle\psi \wedge \langle\varphi\rangle\rho)$
- $c(\langle\varphi\rangle\langle\mathbf{K}\rangle(\psi, \rho)) > c(\langle\mathbf{K}\rangle(\langle\varphi\rangle\psi, \langle\varphi\rangle\rho))$
- $c(\langle\varphi\rangle\langle\mathbf{B}\rangle\psi) > c(\langle\mathbf{B}\rangle(\langle\varphi\rangle\psi \vee \langle\mathbf{K}\rangle(\langle\mathbf{B}\rangle\varphi, \langle\mathbf{B}\rangle\langle\varphi\rangle\psi)))$
- $c(\langle\varphi\rangle\langle\psi\rangle\rho) > c(\langle\varphi\rangle(tr(\langle\psi\rangle\rho)))$

But coming up with a measure $c$ that works is tricky, and is dependent on the specific reduction axioms. For the gritty details involved in coming up with this measure, as well as proving termination for the term rewriting system, see (**?**).

From here, we assume we have this measure $c$. We now have two things left to show:

**Claim 1.** For all $\varphi \in \Gamma^*$, we have $\vdash \varphi \leftrightarrow tr(\varphi)$

*Proof.* By induction on $c(\varphi)$.

**Base Step.** If $\varphi$ is a proposition $p$, then we (trivially) have $\vdash p \leftrightarrow p$.

**Inductive Step.** We consider each possible inductive case, and suppose the claim holds for formulas $\psi$ with smaller $c(\psi)$. The $\neg\varphi$, $\varphi \wedge \psi$, $\mathbf{K}(\varphi, \psi)$, and $\mathbf{B}\varphi$ cases all follow from applying the translation, and then applying inductive hypothesis on the subexpression that results from this.

Here are the rest of the cases. Notice that we apply the inductive hypothesis to terms whose $c$-cost is smaller (this is why we needed the decreasing properties of $c$ before).

$\langle\varphi\rangle p$ **case** We have

$$tr(\langle\varphi\rangle p) = tr(p) = p$$

and so we need to show that

$$\vdash \langle\varphi\rangle p \leftrightarrow p$$

but this holds by our propositional reduction axiom.

$\langle\varphi\rangle\neg\psi$ **case.** We have:

$$\begin{aligned} &\vdash \langle\varphi\rangle\neg\psi \\ &\quad \leftrightarrow \neg\langle\varphi\rangle\psi && \text{(by the reduction axiom)} \\ &\quad \leftrightarrow tr(\neg\langle\varphi\rangle\psi) && \text{(inductive hypothesis)} \\ &\quad = tr(\langle\varphi\rangle\neg\psi) && \text{(by our translation)} \end{aligned}$$

$\langle\varphi\rangle\psi \wedge \rho$ **case.** We have:

$$\begin{aligned} &\vdash \langle\varphi\rangle(\psi \wedge \rho) \\ &\quad \leftrightarrow \langle\varphi\rangle\psi \wedge \langle\varphi\rangle\rho && \text{(by the reduction axiom)} \\ &\quad \leftrightarrow tr(\langle\varphi\rangle\psi \wedge \langle\varphi\rangle\rho) && \text{(inductive hypothesis)} \\ &\quad = tr(\langle\varphi\rangle(\psi \wedge \rho)) && \text{(by our translation)} \end{aligned}$$

$\langle\varphi\rangle\mathbf{K}(\psi, \rho)$ **case.** We have:

$$\begin{aligned} &\vdash \langle\varphi\rangle\mathbf{K}(\psi, \rho) \\ &\quad \leftrightarrow \mathbf{K}(\langle\varphi\rangle\psi, \langle\varphi\rangle\rho) && \text{(by the reduction axiom)} \\ &\quad \leftrightarrow tr(\mathbf{K}(\langle\varphi\rangle\psi, \langle\varphi\rangle\rho)) && \text{(inductive hypothesis)} \\ &\quad = tr(\langle\varphi\rangle\mathbf{K}(\psi, \rho)) && \text{(by our translation)} \end{aligned}$$

$\langle\varphi\rangle\mathbf{B}\psi$ **case.** We have:

$$\begin{aligned} &\vdash \langle\varphi\rangle\mathbf{B}\psi \\ &\quad \leftrightarrow \langle\mathbf{B}\rangle\langle\varphi\rangle\psi \vee \langle\mathbf{K}\rangle(\langle\mathbf{B}\rangle\varphi, \langle\mathbf{B}\rangle\langle\varphi\rangle\psi) \\ &\qquad \text{(by the reduction axiom)} \\ &\quad \leftrightarrow tr(\langle\mathbf{B}\rangle\langle\varphi\rangle\psi \vee \langle\mathbf{K}\rangle(\langle\mathbf{B}\rangle\varphi, \langle\mathbf{B}\rangle\langle\varphi\rangle\psi)) \\ &\qquad \text{(inductive hypothesis)} \\ &\quad = tr(\langle\varphi\rangle\mathbf{B}\psi) \\ &\qquad \text{(by our translation)} \end{aligned}$$

$\langle\varphi\rangle\langle\psi\rangle\rho$ **case.** This case is more interesting. First, notice our translation for this case:

$$tr(\langle\varphi\rangle\langle\psi\rangle\rho) = tr(\langle\varphi\rangle tr(\langle\psi\rangle\rho))$$

That is, we translate the inner expression first, then translate the outer expression. This inner $tr(\langle\psi\rangle\rho)$ is equivalent to some update-operator-free formula $\chi$:

$$\vdash \chi \leftrightarrow tr(\langle\psi\rangle\rho) \leftrightarrow \langle\psi\rangle\rho \tag{1}$$

(This last equivalence follows from our inductive hypothesis, which we can apply because $\langle\psi\rangle\rho$ is a subexpression of $\langle\varphi\rangle\langle\psi\rangle\rho$.)

What about $tr(\langle\varphi\rangle\chi)$? Well, since $\chi$ is update-operator-free, this reduces to our previous inductive cases. So we have

$$\vdash tr(\langle\varphi\rangle\chi) \leftrightarrow \langle\varphi\rangle\chi \tag{2}$$

Putting this all together, we have:

$$\begin{aligned} &\vdash \langle\varphi\rangle\langle\psi\rangle\rho \\ &\quad \leftrightarrow \langle\varphi\rangle\chi && \text{(by (??))} \\ &\quad \leftrightarrow tr(\langle\varphi\rangle\chi) && \text{(by (??))} \\ &\quad \leftrightarrow tr(\langle\varphi\rangle(tr(\langle\psi\rangle\rho))) && \text{(by (??))} \\ &\quad \leftrightarrow tr(\langle\varphi\rangle\langle\psi\rangle\rho) && \text{(by our translation)} \end{aligned}$$

□

**Claim 2.** For all $\varphi \in \Gamma^*$, $\mathcal{N} \models \varphi$.

*Proof.* Let $\varphi \in \Gamma^*$. Recall that we picked $\mathcal{N}$ such that it models

$$\Gamma^{\mathrm{tr}} = \{tr(\varphi) \mid \varphi \in \Gamma^*\}$$

Since $\vdash \varphi \leftrightarrow tr(\varphi)$, and $\mathcal{N} \models tr(\varphi)$, by soundness we have $\mathcal{N} \models \varphi$. □

□

**Theorem 6** (**Completeness**)**.** Suppose we have a complete axiomatization for **K** and **B**. Then the logic of unstable Hebbian learning is completely axiomatized by these laws, plus the above reduction axioms: for all consistent $\Gamma^* \subseteq \mathcal{L}^*$, if $\Gamma^* \models \varphi$ then $\Gamma^* \vdash \varphi$. N

*Proof.* Since our language $\mathcal{L}^*$ has negation, completeness follows from model building in the usual way; this proof is entirely standard.

Suppose contrapositively that $\Gamma^* \nvdash \varphi$. It follows that $\Gamma^* \vdash \neg\varphi$. So $\Gamma^* \cup \{\neg\varphi\}$ is consistent, and by Theorem **??**, we have $\mathcal{N} \in \mathsf{Net}$ such that $\mathcal{N} \models \Gamma^* \cup \{\neg\varphi\}$. But then $\mathcal{N} \models \Gamma^*$ yet $\mathcal{N} \not\models \varphi$, which is what we wanted to show. □

## 5 Running our Code

### Installation Instructions

Our Lean 4 code is contained in the included Zip file. Inside, there is a README.md file with these instructions, and a 'proofs.lean' file with the actual proof code. If you would like to run 'proofs.lean', you will need to install Lean 4 on your own machine (you can just read the source code, too, but Lean has interactive features). The easiest way to do this is to follow the instructions at

https://leanprover.github.io/lean4/doc/quickstart.html

This guide walks you through installing Lean 4 by way of Visual Studio Code, the text editor and enviromnnent we used to develop our code. This will install the most recent nightly release of Lean 4 (this may take a long time).

Lean is in rapid development, but new versions should be backwards-compatible with older ones. We recommend using the version of Lean that is installed. We have checked that our code works with the 2023-03-17 and 2023-08-18 releases.

**System Details.** We developed and ran this code on a personal laptop, i.e., a Dell XPS-15 with a 20 × $12^{\text{th}}$ Gen Intel Core i7 processor (5.0 GHz) and 32GB of RAM. But this is overkill — we recommend using a machine with at least a 1.6GHz processor and 4.0GB RAM.

Lean 4 and Visual Studio Code are available on all major platforms (our code is platform-independent), but we developed in the Kubuntu 22.04 operating system (with KDE Plasma version 5.24.7). We used Visual Studio Code version 1.77.3.

### Running and Reading our Lean Proofs

After you have successfully installed Lean 4, open 'proofs.lean' in Visual Studio Code. When Lean has finished loading the file (this may take a few minutes), you should see text buffer split into two. On the left is the source code, containing defnitions, theorems, proofs, and propositions. On the right you will see the Lean Infoview, which will output all messages, including errors, 'printed' `#eval` statements, and warnings (these can be found in the "All Messages" drop-down menu).

The Lean Infoview may also show the current context of a proof. Figure **??** (on the next page) shows an example of this. The proof on the left shows by induction that if there is a path from $u$ to $v$ in a graph, and a path from $v$ to $w$, then there is a path from $u$ to $w$. If we place our cursor on line 135, then on the right we see the context of the proof at line 135. This shows all variables `u`, `v`, `w`, `g` defined in scope, as well as any assumptions we've made so far $h_1$, $h_2$. Finally, `⊢ hasPath g u w` indicates that our goal at this point is to prove that there is a path from `u` to `w`.

If we move our cursor to line 140 (the end of this proof), the Lean Infoview tells us that there are no goals left. This means that the proof is done and formally checked! Note that proofs that *are not* finished are easily spotted — the keyword **`sorry`** indicates a goal with no proof. (We intend to fill in all **`sorry`**'s by the AAAI-24 author feedback window.)

## 6 Reproducibility Checklist

Most of our answers to the Reproducibility Checklist are straightforward (our contribution is theoretical; we state assumptions and claims formally; we provide proofs and intuitions; we cite other theoretical tools we use). But we we also answered 'yes' to some of the experimental questions as well, namely

10. *All experimental code used to eliminate or disprove claims is included.*
18. *All source code required for conducting and analyzing the experiments is included in a code appendix.*
19. *All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes.*
20. *All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from.*
22. *This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names, and versions of relevant software libraries and frameworks.*

This is because our Lean 4 code doubles as "experimental code used to eliminate or disprove claims." Although we didn't build any counter-models explicitly, Lean's type-checking and resolution system helped us eliminate possible (false) claims in the process of writing our proofs. (Lean 4 helped us avoid many false starts with Theorem **??**.) And when Lean tells us that a proof of a claim is good (there are no goals left), this means that there *aren't* any counter-models for that claim.

We have included all source code in an appendix, and intend to release it under the MIT license. We have included comments in the source code, with each claim referencing the corresponding claim in the paper. And in the previous section (as well as in a README.md file), we detail the system requirements for running this code.

There are two questions that we answered 'no' to:

9. *All theoretical claims are demonstrated empirically to hold.*
26. *The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank).*

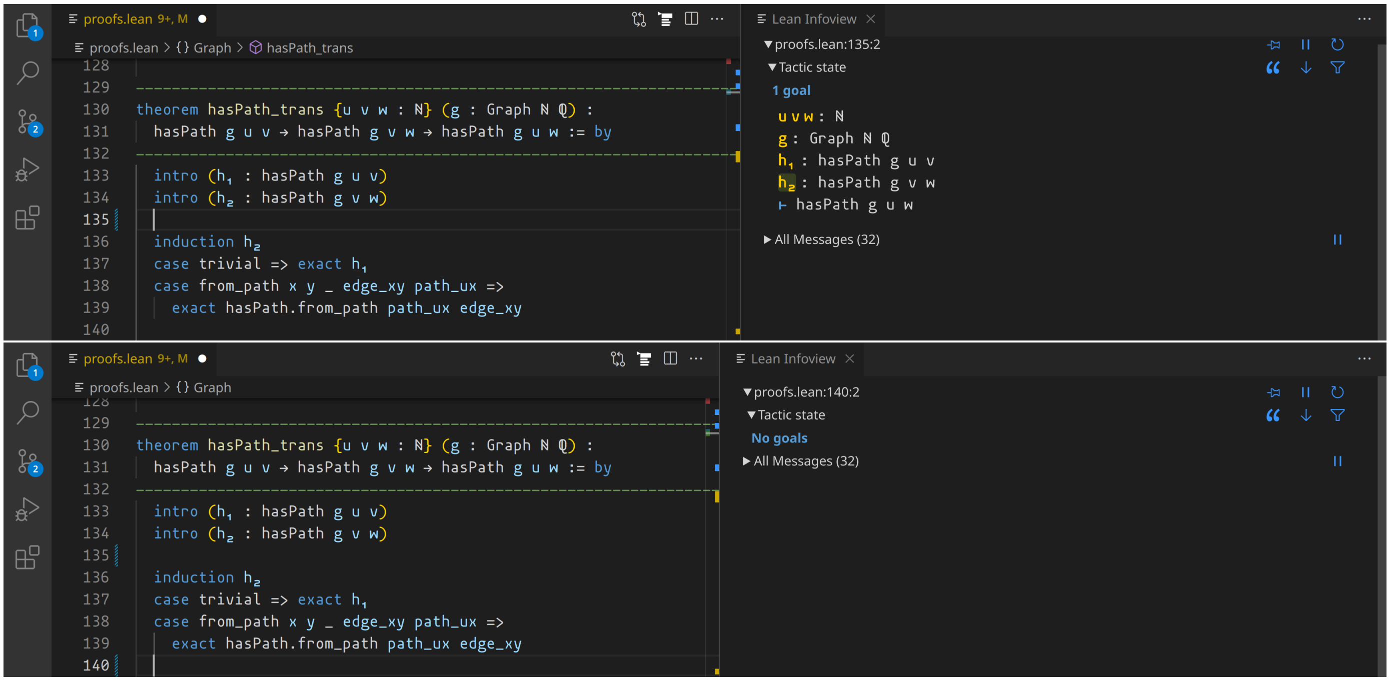
For (9), we unfortunately ran out of time to give an effective example or write code testing our reduction on neural networks out in the wild. Question (26) is not applicable, but at the time of submission the Microsoft CMT system didn't have an option for 'not applicable.'

## References

Baltag, A.; Moss, L. S.; and Solecki, S. 2023. Logics for epistemic actions: completeness, decidability, expressivity. *Logics*, 1(2): 97–147.

Leitgeb, H. 2001. Nonmonotonic reasoning by inhibition nets. *Artificial Intelligence*, 128(1-2): 161–201.

Moura, L. d.; and Ullrich, S. 2021. The Lean 4 theorem prover and programming language. In *Automated Deduction–CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings 28*, 625–635. Springer.



```
128
129
130  theorem hasPath_trans {u v w : ℕ} (g : Graph ℕ ℚ) :
131    hasPath g u v → hasPath g v w → hasPath g u w := by
132
133    intro (h₁ : hasPath g u v)
134    intro (h₂ : hasPath g v w)
135
136    induction h₂
137    case trivial => exact h₁
138    case from_path x y _ edge_xy path_ux =>
139      exact hasPath.from_path path_ux edge_xy
140
```

```
▼proofs.lean:135:2
 ▼Tactic state
  1 goal
   u v w : ℕ
   g : Graph ℕ ℚ
   h₁ : hasPath g u v
   h₂ : hasPath g v w
   ⊢ hasPath g u w
▶All Messages (32)
```

```
128
129
130  theorem hasPath_trans {u v w : ℕ} (g : Graph ℕ ℚ) :
131    hasPath g u v → hasPath g v w → hasPath g u w := by
132
133    intro (h₁ : hasPath g u v)
134    intro (h₂ : hasPath g v w)
135
136    induction h₂
137    case trivial => exact h₁
138    case from_path x y _ edge_xy path_ux =>
139      exact hasPath.from_path path_ux edge_xy
140
```

```
▼proofs.lean:140:2
 ▼Tactic state
  No goals
▶All Messages (32)
```

Figure 1: The Lean Infoview (right), at two different lines in the proof.