Modal Logic Neural Networks

Differentiable Kripke semantics with a *learnable*, *inspectable* accessibility relation

Antonin Sulc[1] Noor Naddour[2]

NeSy 2026 | `sulcantonin/torchmodal` | `pip install torchmodal`

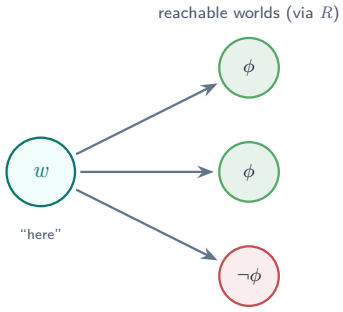
One differentiable engine for modal logic

[1]Lawrence Berkeley National Laboratory

[2]The University of Queensland

BERKELEY LAB U.S. DEPARTMENT OF ENERGY Office of Science The University of Queensland

Modal logic: truth depends on which *world* you are in



$\Box\phi$: *every* reachable is ϕ? **no**

$\Diamond\phi$: *some* reachable is ϕ? **yes**

A proposition need not be flatly true or false; it can hold *in some worlds* and fail *in others*.

$\Box\phi$ **necessity**: true in *every* reachable world

$\Diamond\phi$ **possibility**: true in *some*

Worlds are linked by an **accessibility relation** R, which says which world bears on which.

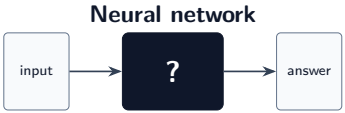
One “must / may”, four questions real systems ask

Reading	A question it answers	Operator
Knowledge	does the controller *know* the tank is empty?	$K_a\phi$
Belief	which negotiating agent should we *believe*?	$B_a\phi$
Time	will the interlock stay engaged *from now on*?	$G\phi,\ \phi\,\mathcal{U}\,\psi$
Obligation	is the system *obliged* to raise an alarm?	$O\phi,\ P\phi$

Each is the same shape, *must / may across reachable worlds*, so **one** logic covers all four, changing only which *frame axioms* the relation R satisfies.

Those axioms are *audited* after training, not imposed: the logic you land in is explored, not assumed.

Black box, graph network, or a contradiction-minimising machine?



pattern in, answer out;
no worlds, no relation, nothing to read

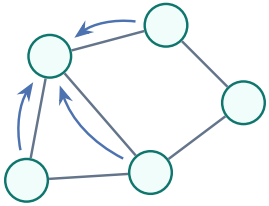
Black box, graph network, or a contradiction-minimising machine?

Neural network

pattern in, answer out;
no worlds, no relation, nothing to read

Graph neural network

learned messages
$m_{ij} = \mathrm{MLP}(h_i, h_j)$



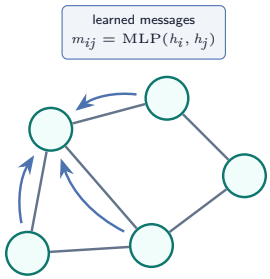
the graph is *given*; it learns messages,
not logic: nothing in it can contradict

Black box, graph network, or a contradiction-minimising machine?

Neural network

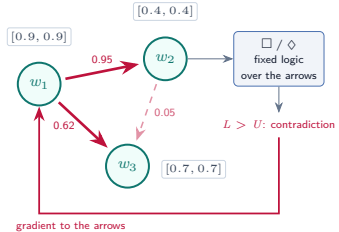
pattern in, answer out;
no worlds, no relation, nothing to read

Graph neural network



the graph is *given*; it learns messages,
not logic: nothing in it can contradict

MLNN

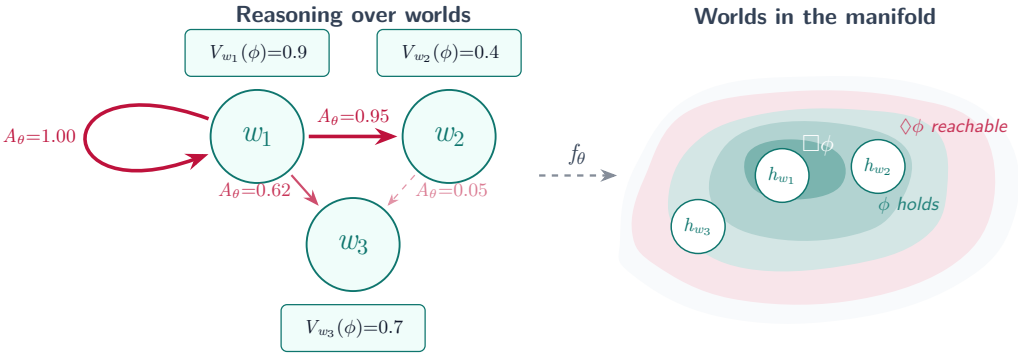


worlds, *learned* arrows, truth as $[L, U]$;
a crossed bound is the training signal

The relation is the unknown.

a black box never shows it, a graph network must be handed it, an MLNN learns it and lets you read it

The whole idea in one picture



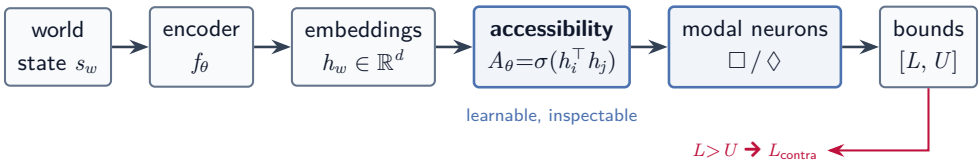
aggregating at w_1 *brackets* V_{w_1}*:* $[\Box\phi, \Diamond\phi]$=[0.45, 0.90]

Fig. 1

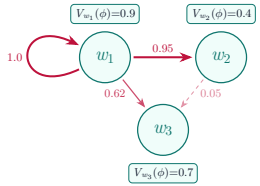
Our answer: make the Kripke triple $\langle W, R, V \rangle$ differentiable

Modal Logic Neural Network (MLNN)

Represent worlds, a valuation V_φ, and the accessibility relation $A_\theta \in [0,1]^{|W|\times|W|}$ *all inside one forward pass*. The rules are yours; of W, V and A_θ, **fix** what you know (deductive) and **learn** the rest, usually A_θ (inductive). Truth is a bounded interval $[L, U]$: a degree, a width of ignorance and, when $L > U$, a contradiction with a *size*.



Worlds, weighted accessibility, and a truth *bracket*

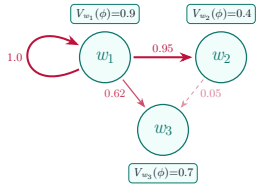


weights $A_\theta \in [0, 1]$ *are* the parameters

- In each world a statement is a range, not one number: how true *at least*, and *at most*.
- Necessity and possibility are new statements about ϕ, judged at w_1 from the worlds it reaches.
- Each gets its own range, computed from ϕ's ranges and the arrows:

Fig. 1, Eq. 2

Worlds, weighted accessibility, and a truth *bracket*



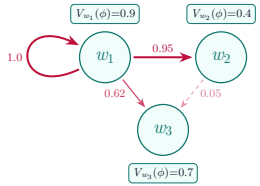
weights $A_\theta \in [0,1]$ *are* the parameters

- In each world a statement is a range, not one number: how true *at least*, and *at most*.
- Necessity and possibility are new statements about ϕ, judged at w_1 from the worlds it reaches.
- Each gets its own range, computed from ϕ's ranges and the arrows:

$$\Box\phi(w_1) = \operatorname*{smin}_{\tau,\, w'} \big[(1-A_\theta)+V\big] \approx \mathbf{0.45}$$

Fig. 1, Eq. 2

Worlds, weighted accessibility, and a truth *bracket*



weights $A_\theta \in [0,1]$ *are* the parameters

- In each world a statement is a range, not one number: how true *at least*, and *at most*.
- Necessity and possibility are new statements about ϕ, judged at w_1 from the worlds it reaches.
- Each gets its own range, computed from ϕ's ranges and the arrows:

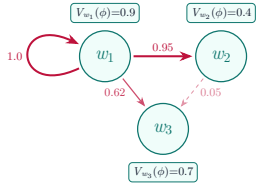
$$\Box\phi(w_1) = \operatorname*{smin}_{\tau,\, w'}\big[(1-A_\theta)+V\big] \approx \mathbf{0.45}$$

$$\Diamond\phi(w_1) = \operatorname*{smax}_{\tau,\, w'}\big[A_\theta+V-1\big] \approx \mathbf{0.90}$$

$\Box$ = weighted $\forall$ ("weakest link") $\Diamond$ = weighted $\exists$ ("evidence scout") neither is the L or U of ϕ itself

Fig. 1, Eq. 2

Worlds, weighted accessibility, and a truth *bracket*



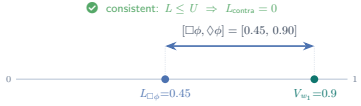
weights $A_\theta \in [0,1]$ *are* the parameters

- In each world a statement is a range, not one number: how true *at least*, and *at most*.
- Necessity and possibility are new statements about ϕ, judged at w_1 from the worlds it reaches.
- Each gets its own range, computed from ϕ's ranges and the arrows:

$$\Box\phi(w_1) = \min_{\tau,\, w'}\big[(1-A_\theta)+V\big] \approx \mathbf{0.45}$$

$$\Diamond\phi(w_1) = \max_{\tau,\, w'}\big[A_\theta+V-1\big] \approx \mathbf{0.90}$$

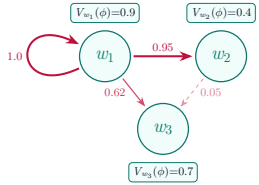
$\Box$ = weighted $\forall$ ("weakest link") $\Diamond$ = weighted $\exists$ ("evidence scout") neither is the L or U of ϕ itself



w_1 sees itself ($A_\theta[w_1, w_1]$=1), so $\Box\phi \le V_{w_1} \le \Diamond\phi$: the sandwich.

Fig. 1, Eq. 2

Worlds, weighted accessibility, and a truth *bracket*



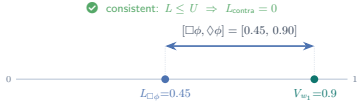
weights $A_\theta \in [0,1]$ *are* the parameters

- In each world a statement is a range, not one number: how true *at least*, and *at most*.
- Necessity and possibility are new statements about ϕ, judged at w_1 from the worlds it reaches.
- Each gets its own range, computed from ϕ's ranges and the arrows:

$$\Box\phi(w_1) = \min_{\tau,\, w'}\big[(1-A_\theta)+V\big] \approx \mathbf{0.45}$$

$$\Diamond\phi(w_1) = \max_{\tau,\, w'}\big[A_\theta+V-1\big] \approx \mathbf{0.90}$$

$\Box$ = weighted $\forall$ ("weakest link") $\Diamond$ = weighted $\exists$ ("evidence scout") neither is the L or U of ϕ itself



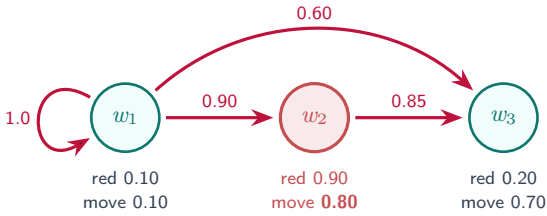
w_1 sees itself ($A_\theta[w_1, w_1]$=1), so $\Box\phi \le V_{w_1} \le \Diamond\phi$: the sandwich.

Any node with $L > U$ ("at least" above "at most") is a contradiction of size $L-U$: the training signal L_{contra}.

Fig. 1, Eq. 2

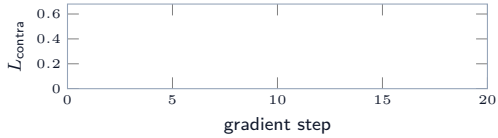
An axiom, a contradiction, and two ways to fix it

On red, never move: $\Box(\text{red} \to \neg\text{move})$ asserted at w_1 with $[L, U]=[1, 1]$, "in every reachable state, red implies not-move". State w_2 is reachable, and there the light is red and the car moves.



arrows: A_θ out of w_1; numbers: truth of each atom

axiom at w_1: asserted $L=1$, computed $U=0.40 \Rightarrow L_{\text{contra}}=0.60$

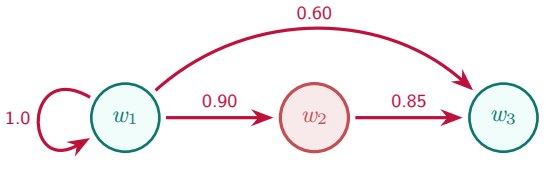


Same machinery, two answers; what you *fixed* decides which one you get.

real numbers: an actual gradient-descent run (torchmodal), not an illustration

An axiom, a contradiction, and two ways to fix it

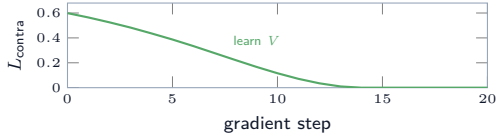
On red, never move: $\Box(\text{red} \rightarrow \neg\text{move})$ asserted at w_1 with $[L, U]=[1, 1]$, "in every reachable state, red implies not-move". State w_2 is reachable, and there the light is red and the car moves.



arrows: A_θ out of w_1; numbers: truth of each atom

axiom at w_1: asserted $L=1$, computed $U=0.40 \Rightarrow L_{\text{contra}}=0.60$

fix A_θ, learn V: move(w_2) $0.80 \rightarrow 0.04 \Rightarrow U=1.00$, $L_{\text{contra}}=0$ (step 13)

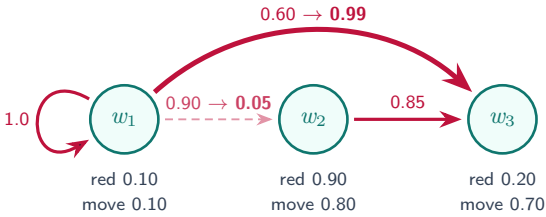


Same machinery, two answers; what you *fixed* decides which one you get.

real numbers: an actual gradient-descent run (torchmodal), not an illustration

An axiom, a contradiction, and two ways to fix it

On red, never move: $\Box(\text{red} \rightarrow \neg\text{move})$ asserted at w_1 with $[L, U]=[1, 1]$, "in every reachable state, red implies not-move". State w_2 is reachable, and there the light is red and the car moves.

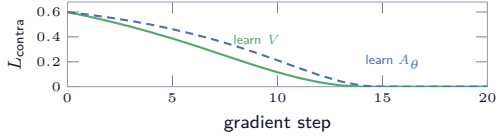


arrows: A_θ out of w_1; numbers: truth of each atom

axiom at w_1: asserted $L{=}1$, computed $U{=}0.40 \Rightarrow L_{\text{contra}}{=}0.60$

fix A_θ, learn V: move(w_2) $0.80 \rightarrow 0.04 \Rightarrow U{=}1.00$, $L_{\text{contra}}{=}0$ (step 13)

fix V, learn A_θ: $A_\theta(w_1, w_2)$ $0.90 \rightarrow 0.05$, $A_\theta(w_1, w_3)$ $0.60 \rightarrow 0.99 \Rightarrow U{=}1.00$, $L_{\text{contra}}{=}0$ (step 14)



Same machinery, two answers; what you *fixed* decides which one you get.

real numbers: an actual gradient-descent run (torchmodal), not an illustration

Operators & frame axioms: the two dials the user sets

The operators (soft, sound bounds):

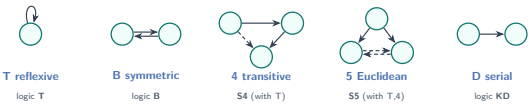
$$\Box\phi(w) = \operatorname*{smin}_{\tau,\, w'} \big[(1 - A_\theta) + V_{w'}\big] \qquad \textit{weakest link}$$

$$\Diamond\phi(w) = \operatorname*{smax}_{\tau,\, w'} \big[A_\theta + V_{w'} - 1\big] \qquad \textit{evidence scout}$$

$$(\phi\,\mathcal{U}\,\psi)_t = \psi_t \vee \big(\phi_t \wedge (\phi\,\mathcal{U}\,\psi)_{t+1}\big) \qquad \textit{until}$$

One engine computes all three; only the *frame* the relation satisfies changes the reading.

Example axioms A_θ may satisfy:



They are *audited* after training, not imposed; which subset holds *is* the modality.

Knowledge $K_a\phi$	**T 4 5** (S5)	**T**: the real world is among the candidates ⇒ what is known is *true*
Belief $B_a\phi$	**D 4 5** (KD45)	no **T**: the real world may be excluded ⇒ a belief can be *wrong*, yet consistent (D) and introspective (4, 5)
Obligation $O\phi$	**D** (KD)	an ideal world always exists ⇒ duties are consistent; *this* world need not be ideal
Time $G\phi$	**4** (+D)	later than later is later

Four experiments: the object computed *is* the object read

Modality	Task	What A_θ reads out as
Doxastic B_a	LLM-swarm saboteur	a **trust matrix**
Deontic O/P	C-MAPSS turbofan wear	a **regime + safety** embedding
Temporal $\mathcal{U}$	ROCStories ordering	a **precedence** order
CSP $\square$	graph colouring	a **recovered constraint graph**

Each is one instance of the *same* loss $L_{\text{total}} = L_{\text{task}} + \beta L_{\text{contra}}$: the relation A_θ is trained by L_{contra} alone in all four, and three of the four have no task loss at all ($L_{\text{task}}{=}0$).

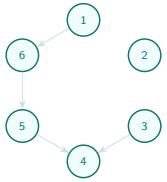
Accuracy is competitive with strong baselines, but the *contribution is the auditable structure they do not express.*

Doxastic: who should we believe? A trust matrix finds the saboteur

1. Six chatbots negotiate a budget; one is secretly told to derail it.
2. A frozen NLI model scores each message against "the budget is fair": $e = P(\text{entailment})$, $c = P(\text{contradiction})$.
3. Per agent $L =$ top e, $U = 1-$ top c. Switching sides $\Rightarrow L > U$: the others stop trusting it.

✓ **honest agent:** "I remain committed to the agreed budget…", every round $\Rightarrow$ trust **kept**

× **saboteur:** "I concur…" then "a minimum of \$75,000…" $\Rightarrow L > U$, trust **drained**

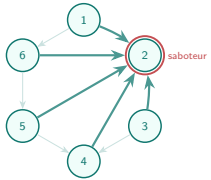


Doxastic: who should we believe? A trust matrix finds the saboteur

1. Six chatbots negotiate a budget; one is secretly told to derail it.
2. A frozen NLI model scores each message against "the budget is fair": $e = P(\text{entailment})$, $c = P(\text{contradiction})$.
3. Per agent $L =$ top e, $U = 1-$ top c. Switching sides $\Rightarrow L > U$: the others stop trusting it.

✓ **honest agent:** "I remain committed to the agreed budget…", every round ⇒ trust **kept**

× **saboteur:** "I concur…" then "a minimum of $75,000…" ⇒ $L > U$, trust **drained**



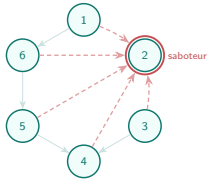
arrows into agent 2: how much the others trust it

Doxastic: who should we believe? A trust matrix finds the saboteur

1. Six chatbots negotiate a budget; one is secretly told to derail it.
2. A frozen NLI model scores each message against "the budget is fair": $e = P(\text{entailment})$, $c = P(\text{contradiction})$.
3. Per agent $L =$ top e, $U = 1-$ top c. Switching sides $\Rightarrow L > U$: the others stop trusting it.

✓ **honest agent:** "I remain committed to the agreed budget…", every round ⇒ trust **kept**

× **saboteur:** "I concur…" then "a minimum of $75,000…" ⇒ $L > U$, trust **drained**

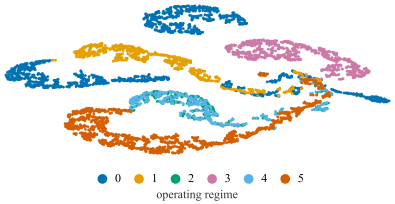


the saboteur's trust drains to zero

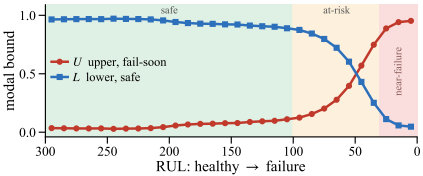
- **Caught in 12 of 15** subtle runs (LLM judge 3 of 15); **15 of 15** under planted text (judge 5 of 15).
- Most of the catching is the reader's; the logic adds a trust matrix you can *read*, immune to planted text.
- ▶ **Live demo at the poster session**: watch the trust matrix drain as the agents talk.

Deontic: obligation and permission, read off jet-engine sensors

- NASA's C-MAPSS: 260 jet engines run from new to failure, 24 sensor readings per cycle.
- Two rules: an engine *must* be safe (obligation); it *may* be about to fail (permission); safe := not about to fail, so never both.
- Read out per cycle: $L_{\Box\,\text{safe}}$ and $U_{\Diamond\,\text{fail-soon}}$; their conflict is the training loss, the only signal that shapes the arrows.



engine states laid out by the network, coloured by regime: the **six regimes** stay apart, unnamed



along an engine's life: "safe" falls and "about to fail" rises as it wears

Temporal (ROCStories): one *Until* formula scores an order

Each sentence is a world; a learned asymmetric head gives $A_\theta(s_i, s_j)$ = "s_j plausibly follows s_i". One *Until* at the first world scores an order σ (all $5!{=}120$ are scored):

$$(\mathit{follows}\,\mathcal{U}\ \mathit{last})_t = \mathit{last}_t \ \vee\ \big(\mathit{follows}_t \wedge (\mathit{follows}\,\mathcal{U}\ \mathit{last})_{t+1}\big), \qquad \mathit{follows}_t = A_\theta(\sigma_{t-1}, \sigma_t)$$

a real test story, true order

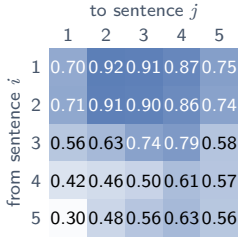
1 Kelly was playing her new Mario game.

2 She had been playing it for weeks.

3 She was playing for so long without beating the level.

4 Finally she beat the last level.

5 Kelly was so happy to finally beat it.



the learned $A_\theta(s_i, s_j)$ for this story

Training pushes the true order above every scramble by a margin. Held-out (mpnet): pairwise 0.82, exact 0.33, level with a prompted 7B LLM at $\sim 10^5$ trainable weights; a symmetric kernel collapses to 0.57.

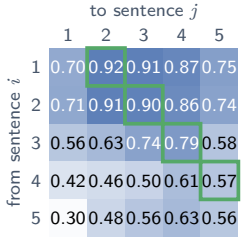
Temporal (ROCStories): one *Until* formula scores an order

Each sentence is a world; a learned asymmetric head gives $A_\theta(s_i, s_j)$ = "s_j plausibly follows s_i". One *Until* at the first world scores an order σ (all $5!{=}120$ are scored):

$$(\mathit{follows}\,\mathcal{U}\ \mathit{last})_t = \mathit{last}_t \ \vee\ \big(\mathit{follows}_t \wedge (\mathit{follows}\,\mathcal{U}\ \mathit{last})_{t+1}\big), \qquad \mathit{follows}_t = A_\theta(\sigma_{t-1}, \sigma_t)$$

a real test story, true order

1 Kelly was playing her new Mario game.
2 She had been playing it for weeks.
3 She was playing for so long without beating the level.
4 Finally she beat the last level.
5 Kelly was so happy to finally beat it.



the learned $A_\theta(s_i, s_j)$ for this story
true order $1{\to}2{\to}3{\to}4{\to}5$**:** every link strong,
$U{=}0.61$, top of all 120 orders

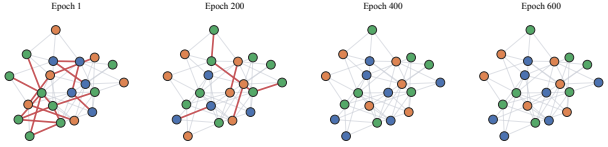
Training pushes the true order above every scramble by a margin. Held-out (mpnet): pairwise 0.82, exact 0.33, level with a prompted 7B LLM at $\sim 10^5$ trainable weights; a symmetric kernel collapses to 0.57.

CSP (graph colouring): the axiom, a contradiction, and a solve

Adjacency *is* accessibility: "neighbours differ" is exactly $\bigwedge_{c=1}^{k}(p_c \to \neg\Diamond p_c)$, "if I am colour c, no adjacent node is c."

rule $p_c \to \neg\Diamond p_c$ violated by $p_c \wedge \Diamond p_c$: $L > U$

- **Solve** (map given, colours learned): **29 of 30**, tying an RRN-style GNN.
- Non-modal ablation (operator $\to$ quadratic penalty on neighbours): 18 of 30 (p=0.001).
- **Learn** (map hidden, 60 valid colourings given): find the arrows under which the axiom holds for every colouring.
- Recovers **every edge (AUC 1.000)**: the relation you read *is* the constraint graph.



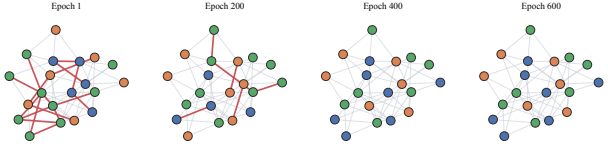
red monochromatic edges are per-node contradictions; they drain to a proper 3-colouring.

CSP (graph colouring): the axiom, a contradiction, and a solve

Adjacency *is* accessibility: "neighbours differ" is exactly $\bigwedge_{c=1}^{k}(p_c \to \neg\Diamond p_c)$, "if I am colour c, no adjacent node is c."

recoloured: $L \leq U$ (consistent)

- **Solve** (map given, colours learned): **29 of 30**, tying an RRN-style GNN.
- Non-modal ablation (operator $\to$ quadratic penalty on neighbours): 18 of 30 (p=0.001).
- **Learn** (map hidden, 60 valid colourings given): find the arrows under which the axiom holds for every colouring.
- Recovers **every edge (AUC 1.000)**: the relation you read *is* the constraint graph.



red monochromatic edges are per-node contradictions; they drain to a proper 3-colouring.

The relation is the unknown.

We *learn* it, *bound* it, and let you *read* it.

`github.com/sulcantonin/torchmodal` | `pip install torchmodal`

Takeaways

1. **Differentiable Kripke semantics**: $\langle W, A_\theta, V_\varphi \rangle$ in one forward pass; $\Box/\Diamond$ as sound, bounded aggregators over a learned A_θ.
2. **One engine, four modalities**; the modality is fixed by frame axioms.
3. **Inspectability is built into inference**, verified exactly on graph colouring.
4. torchmodal: open-source PyTorch library; every number reproducible from committed artefacts.

Acknowledgment: Thanks Innovatrics for supporting my work.

Thank you; questions welcome.

Backup slides

Backup: grammar & Łukasiewicz connectives

$$\phi ::= p \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \phi \to \psi \mid \Box\phi \mid \Diamond\phi \mid \phi \,\mathcal{U}\, \psi$$

Łukasiewicz (real-valued) connectives:

$$\begin{aligned} \neg\phi &= 1 - \phi \\ \phi \wedge \psi &= \max(0, \phi + \psi - 1) \\ \phi \vee \psi &= \min(1, \phi + \psi) \\ \phi \to \psi &= \min(1, 1 - \phi + \psi) \end{aligned}$$

Each formula = truth bounds $[L_\phi, U_\phi]$ per world, on a DAG of subformulae. Inference propagates bounds by the LNN upward/downward algorithm.

Crisp $\{0, 1\}$ relations are the special case: a classical LNN is the zero-one MLNN, so **MLNN generalises LNNs**.

Backup: the three soft aggregators

$$\begin{aligned}
\operatorname{smin}_\tau(x) &= -\tau \log \textstyle\sum_i e^{-x_i/\tau} && \le \min(x) && \text{(sound lower bound)} \\
\operatorname{smax}_\tau(x) &= \tau \log \textstyle\sum_i e^{x_i/\tau} && \ge \max(x) && \text{(sound upper bound)} \\
\text{conv-pool}_\tau(x, z) &= \textstyle\sum_i w_i x_i, \quad w = \operatorname{softmax}(z/\tau) && \in [\min, \max] && \text{(convex pool)}
\end{aligned}$$

- "smin/smax" name log-sum-exp aggregators, *not* the normalising softmax.
- conv-pool takes the bounding direction of its use: $z{=}{-}x$ upper-bounds $\min$; $z{=}x$ lower-bounds $\max$.
- All four bounds are clamped to $[0, 1]$ (smin can exceed 1 on a single-term bracket).
- Default $\tau = 0.1$; the fan-in rule $\tau = O(1/\log n)$ sets the rest.

Backup: the necessity & possibility neurons

With $\bar{A}_{\theta\, w,w'} := 1 - \tilde{A}_{w,w'}$:

$$L_{\Box\phi,w} = \underset{\tau,\, w'}{\operatorname{smin}} \left(\bar{A}_{\theta\, w,w'} + L_{\phi,w'}\right), \qquad U_{\Box\phi,w} = \underset{\tau,\, w'}{\text{conv-pool}} \left(\bar{A}_{\theta\, w,w'} + U_{\phi,w'}\right),$$

$$L_{\Diamond\phi,w} = \underset{\tau,\, w'}{\text{conv-pool}} \left(\tilde{A}_{w,w'} + L_{\phi,w'} - 1\right), \qquad U_{\Diamond\phi,w} = \underset{\tau,\, w'}{\operatorname{smax}} \left(\tilde{A}_{w,w'} + U_{\phi,w'} - 1\right).$$

- The weight enters through the implication $1 - \tilde{A}_{ij} + L_{\phi,w_j}$ inside smin_τ: at $\tilde{A}_{ij} \approx 1$ the truth value fully participates; at ≈ 0 it is removed.
- Duality $\Diamond\phi \equiv \neg\Box\neg\phi$ holds via $\operatorname{smax}_\tau(x) = 1 - \operatorname{smin}_\tau(1 - x)$, so an implementation needs only necessity.

Backup: “isn’t this just a GNN?”

A Kripke frame *is* a graph and $\Box/\Diamond$ are $\min / \max$ over neighbours, so MLNN’s forward pass *is* a GNN. Three things it pins down that a generic GNN leaves free:

	generic GNN	**MLNN**
objective	fit *labels* (BCE / MSE)	**minimise contradiction** $\sum \mathrm{relu}(L-U)$ (from LNN); no task labels
the relation	learnable too (attention, NRI), but to serve the task	learned to **satisfy axioms**: *means* accessibility, recovers the true graph (AUC 1.0)
operators	free MLPs; one point estimate	fixed, **sound** $\Box/\Diamond$ bounds $[L, U]$; $L>U$ localises a contradiction

On a *given* graph an RRN-style GNN ties us $(29/30)$; MLNN earns its keep where the graph is **unknown**, or you need **bounds, an audit, or no labels**.

A GNN whose inductive bias is *pinned to modal logic*; the restriction is the product.

Related work; §5.4

Backup: doxastic per-backbone detection (5×3)

	Simple		Subtle		Inject	
Backbone	MLNN	judge	MLNN	judge	MLNN	judge
llama3	3	3	**3**	0	**3**	0
mistral	3	3	**2**	0	**3**	0
qwen2.5:7b	3	3	**3**	0	**3**	1
qwen2.5:14b	3	3	3	3	3	3
gemma:7b	3	3	**1**	0	**3**	1
Total	15	15	**12**	3	**15**	5

The separation is concentrated on 7 to 8B backbones; qwen2.5:14b is at ceiling for *both* methods. The supported claim is conditional on scale, and falsifiable at 30B+.

Backup: C-MAPSS: two heads, disjoint gradients

Each (engine, cycle) is a world; both heads read the 24-dim sensor vector only.

- Accessibility $f_\theta : \mathbb{R}^{24} \to \mathbb{R}^{16}$ (2642 params); $A_e = \sigma(s\, z_e z_e^\top + b)$ block-per-engine (*exact*, within-engine).
- Valuation $V_\phi(s) = \sigma(\mathrm{MLP}(s))$ (1665 params), $V_{\text{safe}} = 1 - V_\phi$.

The two losses we optimise, one per head:

$$\mathcal{L}_V = \mathrm{BCE}(V_\phi, \mathbb{1}[\mathrm{RUL}{<}30]) \;\; (\to V_\phi \text{ only}), \qquad \mathcal{L}_{\text{deon}} \;(\to A_\theta \text{ only},\; V \text{ detached})$$

$$\mathcal{L}_{\text{deon}} = \underbrace{(1 - U_{\Diamond\text{fail-soon}})^2}_{\mathrm{RUL}<30} + \underbrace{U^2_{\Diamond\text{fail-soon}} + (1 - L_{\Box\text{safe}})^2}_{\mathrm{RUL}>100} \qquad \text{read out: } L_{\Box\text{safe}},\; U_{\Diamond\text{fail-soon}}$$

The at-risk middle (30 to 100) carries *no* term, shaped only through A_θ. RUL supervises V_ϕ in training but is *never read at inference*.

Backup: ROCStories: the pairwise-head control

ordering model	encoder	cl pair	cl exact
cosine (no head)	mpnet	0.495	0.014
pairwise head (supervised)	mpnet	0.809	0.213
MLNN (Until, ours)	mpnet	**0.819**	**0.327**
qwen2.5-7b (prompted)	n/a	0.854	0.330

The supervised pairwise head ($\sim 3\times$ our parameters) *matches* MLNN on local precedence (pair) yet recovers far fewer full orders (exact): its ten pairwise verdicts per story are mutually inconsistent, while the *Until* recurrence enforces one globally consistent chain by construction. That global consistency is what the modal structure buys.

Backup: torchmodal (1/3): the operators, and the traffic-light axiom of slide 8

```
import torch, torchmodal.functional as F

A = torch.tensor([[1.0, .95, .62], [0, 1.0, .05], [0, 0, 1.0]])   # the arrows of slide 5
V = torch.tensor([.9, .4, .7])                                    # phi's truth per world
prop = torch.stack([V, V], -1)                      # bounds [L, U] per world, shape (W, 2)
box  = F.necessity(prop, A, tau=0.1)                # [0.45, 0.46] at w1: the weakest link
dia  = F.possibility(prop, A, tau=0.1)              # [0.90, 0.90] at w1: the best evidence

A = torch.tensor([[1.0, .90, .60], [0, 1.0, .85], [0, 0, 1.0]])   # slide 8: three states
red, move = torch.tensor([.1, .9, .2]), torch.tensor([.1, .8, .7])
psi    = F.implication(red, F.negation(move))     # red -> not move, per world
box    = F.necessity(torch.stack([psi, psi], -1), A, tau=0.1)
contra = torch.relu(1 - box[0, 1])                # asserted L=1 above computed U: 0.60
contra.backward()                                 # Adam on A, or on move, clears it (slide 8)
```

Every operator works on bounds and is differentiable end to end: `F.necessity`, `F.possibility`, `F.until`, the Łukasiewicz connectives, and `F.contradiction`.

Eq. 2 and 3

Backup: torchmodal (2/3): the four experiments are each a few lines

```
# graph colouring:  p_c -> not <>p_c  per node; the adjacency IS the accessibility
viol = (1 - F.implication(pc, F.negation(F.possibility(pc, adj)))).clamp(min=0)

# doxastic trust:  maximise []_i(coherent) over a learnable trust matrix T
loss = ((1 - F.necessity(consistent, T)) ** 2).mean()

# deontic (C-MAPSS):  fail-soon MAY hold late, safe MUST hold early
U_fail = F.possibility(fail, A)[:, 1];  L_safe = F.necessity(safe, A)[:, 0]
loss = ((1 - U_fail) ** 2)[rul < 30].mean() + (U_fail ** 2 + (1 - L_safe) ** 2)[rul > 100].mean()

# temporal (ROCStories):  one Until at the first world scores a sentence order
score = F.until(follows, is_last, chain, tau=0.1)[0]
```

Same engine, four readings: only the relation (`adj`, `T`, `A`, `chain`) and the formula change. All four run in the released examples.

Backup: torchmodal (3/3): losses, regularisers and models

```
from torchmodal import ContradictionLoss, AxiomRegularization, KripkeModel
from torchmodal import MultiAgentKripke, TemporalOperator

F.contradiction(bounds)                  # sum of relu(L - U): the training signal
ContradictionLoss()(bounds)              # the same as a module (mean / squared)
AxiomRegularization(reflexivity=1., symmetry=1.)(A)   # soft T / B / 4

KripkeModel(num_worlds, accessibility)  # Fixed-, Learnable- or MetricAccessibility
MultiAgentKripke(num_agents=6)           # the trust model of the saboteur experiment
TemporalOperator(num_steps=5)            # G / F / Until over a chain of worlds
```

`pip install torchmodal`. The library ships the operators, the dense, metric and fixed accessibility parameterisations, the losses above, and the four examples with the scripts that regenerate every number in this talk. `sulcantonin/torchmodal`

asulc@lbl.gov

github.com/sulcantonin/torchmodal

Learn the relation. Bound the truth. Read the artefact.