Synthesizing Antecedent and Defeater Structure in Exact Neural Causal Laws by Direct Editing and Training

Davide Bragetti
Anthropic Fellow
ORCID: 0009-0008-7645-8241
bragettidavide@gmail.com

***Abstract*—For a float32 ReLU classifier under a declared binary32 semantics, the classes it assigns to all interventions on a few pixels of a fixed image form a finite object, its finite intervention law; for one target class it is a Boolean function that can be enumerated exactly. We study how a prescribed structure of such a law can be synthesized. On a 784–128–10 MNIST classifier and one anchor image, the target is a predicate on the complete one-pixel and pixel-pair laws: no single-pixel change reaches class t before; afterwards exactly one designated pixel does, above a threshold, and a second pixel blocks it at high values. A closed-form rewrite of one hidden unit, inactive on all 70,000 MNIST images, introduces the predicate for all 9 non-anchor classes and changes no MNIST prediction. Ordinary training with one distinct crafted example, appended to 15–30 declared batches, introduces it for 8 classes, with the blocking pixel and schedule found by exhaustive search and a declared rule. Against a matched control that reproduces the baseline bit for bit, restoring the first-layer weights of either pixel shows that the training change is necessary for both the route and the block; a sham insertion indicates a target-specific response of the dataset for 6 of these classes. For classes 5 and 7, a one-unit edit fitted by a linear program and the trained model have identical laws of the target class on all one-pixel and pair interventions, although their parameter changes differ by orders of magnitude in size and support. All positive results are exhaustive binary32 checks of saved models.**

I. Introduction

A trained classifier evaluated under a declared finite-precision semantics is a finite deterministic function. Fix an image and a few free pixels: the class the network outputs for every assignment of those pixels is then completely determined. We call this finite object the network's *finite intervention law* on that domain, or its *causal law* for short, in the restricted sense made precise in section II: the free pixels are set by intervention, the network is fixed, and nothing is claimed about how images arise. Restricted to the event "the predicted class is t", the law is a Boolean function that can be enumerated and stored exactly. On it, statements of the form "this pixel condition suffices for class t" or "this second condition blocks it" are exact set relations. They are the sufficiency and defeat relations familiar from the logic of conditions [1, 2], evaluated on a finite law.

This paper asks how such a structure can be *produced*:

> *Can a formally specified sufficient-condition-with-defeater structure be introduced into the exact finite intervention law of a trained network, by a direct parameter edit and by a training run on one crafted example, and do the two mechanisms realize the same structure, or even the same law?*

The structure is a predicate P on the law; introducing it means P is false on the law before the change and true after it, with the image, pixels, class and semantics fixed. Acceptance is exhaustive evaluation of every state P quantifies over. *Mechanism D* writes the structure directly into one hidden unit. *Mechanism T* lets the baseline's own training run write it, with one distinct crafted example appended to declared batches. The direct route is the constructive reference; the training route is the hard case, because the training run, rather than the experimenter, determines which parameters change.

Results and their status. Table I lists the results. "Proved" marks mathematical statements; "exhaustively verified" marks exact facts about saved models, decided on every state of the declared domain under the declared semantics; "conjectured" marks the one mechanism we propose but do not prove.

What is known. Direct weight edits that take over low-impact units are known [3], as are ReLU indicator-and-AND constructions [4]. One crafted training example can implant a behaviour [5, 6]; one poison provably backdoors linear models and, with a random poison direction, wide bias-free ReLU networks under gradient flow, and an input direction unused by the benign data is the favourable case, with guarantees proved for linear models [7]. Our routing pixel is such a direction; our setting (biases, width 128, Adam, a repeated example) lies outside the hypotheses of those theorems, so our result is an instance of the same mechanism class, not a consequence of them. We claim none of these ingredients. What we add is a synthesis target that is a predicate on a complete finite intervention law, its realization by both mechanisms with exhaustive binary32 acceptance and attribution against a matched control, and the observation that the two mechanisms can meet in the identical law of the target class along parameter changes that differ in size and support by orders of magnitude.

Large language model (LLM) tools were used extensively in this work: for formalization support, experiment and verification code, literature search, drafting and internal review. They are not authors. The research direction, the hypotheses and the decision to accept or reject every result are the author's own; the author takes full responsibility for the content.

TABLE I
RESULTS AND THEIR EPISTEMIC STATUS. ALL VERIFIED STATEMENTS ARE ABOUT ONE TRAINED NETWORK, ONE ANCHOR IMAGE AND THE DECLARED DOMAINS; NONE IS A RATE OVER NETWORKS OR IMAGES.

	result	status	where
R1	finite intervention law, exact Boolean form, introduction of a predicate	definitions; proved	section II
R2	direct synthesis: P_1 for 9/9 classes, 0 of 70,000 predictions changed	proved (locality) + exhaustively verified	section III
R3	training synthesis with one distinct example: P_1 for 8/9 classes	exhaustively verified	section VI
R4	matched-control attribution of both components (T-R2), 8/9	exhaustively verified	section VII
	target-specificity (sham), 6/8; dataset-response mechanism	verified; conjectured	section VII
R5	same law from both mechanisms (T-R3), classes {5,7}	exhaustively verified	section IX
R6	pair-domain law does not fix the predicates; exact witness	proved (elementary) + exhaustively verified	section X

II. EXACT FINITE INTERVENTION LAWS AND THE TARGET

Network and semantics. The baseline θ_0 is a float32 784–128–10 ReLU classifier for MNIST (101,770 parameters, file SHA-256 prefix `4cc8a11f`, test accuracy 97.3%), trained by the trainer described in section V. It is not binarized and not quantized. A uint8 image u is mapped to $x_i = \mathrm{fl}(u_i \cdot \mathrm{fl}(1/255))$; each pre-activation is the left-to-right binary32 fold of the products followed by the bias, with round-to-nearest-even and no fused multiply–add; $\mathrm{ReLU}(a) = a$ if $a > +0$ and $+0$ otherwise; the predicted class $f_\theta(u)$ is the first strict maximum of the ten logits. We write this semantics $\mathcal{S}$; every statement about a law refers to it.

Finite intervention laws. Fix the anchor x, MNIST test image 2,450 of class 3. For a support S of pixels and $v \in \{0,\dots,255\}^S$, $I_x(S,v)$ is x with the pixels of S set to v; D_S is the set of these $256^{|S|}$ images. The *class law* on S is f_θ restricted to D_S, and the *law of class* t is

$$\Lambda^x_{t,S}(v) = \mathbf{1}[f_\theta(I_x(S,v)) = t], \qquad R^x_{t,S} = \{v : \Lambda^x_{t,S}(v) = 1\}.$$

In what sense the law is causal. We use "causal" in one restricted, computational sense, and the formal object is always the finite intervention law above. (i) The manipulated variables are input pixels, which have no parents in the network's computation graph, so setting them is a well-defined hard intervention on the network viewed as a deterministic computational model [8]. (ii) The network, its parameters and its semantics are fixed. (iii) The observable is one event, the predicted class being t. (iv) The domain is finite and declared. Within these four conditions, $\Lambda^x_{t,S}$ is exactly the effect of the interventions on the event. Nothing is claimed about the process that generates images, no structural causal model of data is identified, and nothing is discovered from observational data; changing the network, the semantics, the anchor or the domain gives a different law.

The Boolean object is the law, not the network. Encode each free pixel by its exact 8-bit binary representation, $\mathrm{enc} : \{0,\dots,255\}^S \to \{0,1\}^{8|S|}$, a bijection. The Booleanized law $\tilde{\Lambda}^x_{t,S} = \Lambda^x_{t,S} \circ \mathrm{enc}^{-1}$ is then a Boolean function of $8|S|$ variables that agrees with $\Lambda^x_{t,S}$ on every state, so every query on one (region, model count, sufficiency of a condition, defeat) is the same query on the other. A reduced ordered binary decision diagram stores it canonically for a fixed variable order [9]. This Booleanization concerns the finite law only: the network is a float32 ReLU network, it is never compiled into a circuit, and it is not a binarized neural network.

Support horizons. $H_k(x)$ is the set of images that differ from x in at most k pixels, $R^{(k)}_t(x) = \{x' \in H_k(x) : f_\theta(x') = t\}$, and the minimum support of t is $k^*(x,t) = \min\{k : R^{(k)}_t(x) \neq \emptyset\}$. The one-pixel horizon has $784 \times 256 = 200{,}704$ states; at θ_0 all of them are class 3.

Sufficiency and defeat. A condition A on the free pixels is *sufficient* for t on a domain D if every state of D satisfying A is in the region: $[\![A]\!] \cap D \subseteq R$. A target state outside every known sufficient condition is an *alternative-antecedent witness*. A context C is a *conjunctive defeater* of a condition A trusted to be sufficient if $\emptyset \neq [\![A \wedge C]\!] \not\subseteq R$, and a *strong* (blocking) one if $[\![A \wedge C]\!] \cap R = \emptyset$. A *background defeater* of a statement certified on a domain D_0 is an extension $D_1 \supset D_0$ on which the statement fails. These are exact set relations on a finite law. The shapes are classical (plural sufficient conditions, defeasible conditions); reading them as paradoxes of partial causal knowledge is an interpretation that motivates the study, and every result below holds if that word is dropped.

Definition 1 (Target predicates)**.** *Let pixel* 0 *be the lowest-index pixel that is* 0 *in all 70,000 MNIST images and in the anchor. For a class* $t \neq 3$ *and the law of parameters* θ*:* BASE*:* $R^{(1)}_t(x) = \emptyset$ *at the reference parameters;* E1*:* $k^*(x,t) = 1$ *and every state of* $R^{(1)}_t(x)$ *changes pixel* 0*;* E2*:* $R^x_{t,\{0\}} = \{a,\dots,255\}$ *for some* a*, so* $A = [x_0 \geq a]$ *is exactly sufficient on pixel* 0*'s law;* E3(j)*: for a pixel* $j \neq 0$ *and the least* c *with* $[\![A \wedge [x_j \geq c]]\!] \cap R^x_{t,\{0,j\}} = \emptyset$*,* $c \leq 255$*, while* $[\![A \wedge [x_j = u_j]]\!] \subseteq R^x_{t,\{0,j\}}$*.* $P^{\mathrm{core}}_1(t) = \mathrm{E1} \wedge \mathrm{E2}$*, and* $P_1(t;0,j) = P^{\mathrm{core}}_1(t) \wedge \mathrm{E3}(j)$*. We call* a *the route threshold and* c *the block threshold. For a pair* S*,* $P_{\mathrm{hor}}(t;S)$ *holds if* $R^{(1)}_t(x) = \emptyset$ *and* $R^x_{t,S} \neq \emptyset$*: the minimum support of* t *is certified to be* 2*, with witness support* S*.*

Definition 2 (Introduction)**.** *For a predicate* P *with its reference fixed (anchor, pixels, class, semantics),* $\theta_0 \mapsto \theta_1$ introduces P *if* $P(\Lambda_{\theta_0}) = 0$ *and* $P(\Lambda_{\theta_1}) = 1$*.*

Proposition 1 (Formal reading)**.** *(1) Under BASE, E1 makes every state of* $R^{(1)}_t(x)$ *an alternative-antecedent witness relative to complete knowledge of the one-pixel law of the reference parameters. (2) If E2 and E3*(j) *hold, every state of* $[\![A \wedge [x_j \geq c]]\!]$ *lies in* $D_{\{0,j\}} \setminus D_{\{0\}}$*: the strong conjunctive*

defeat of A is a background defeat of the domain on which A is certified sufficient, and no conjunctive defeater of A exists on $D_{\{0\}}$ itself. (3) If $P_{\mathrm{hor}}(t;S)$ holds, every state of $R^x_{t,S}$ is an alternative-antecedent witness relative to complete one-pixel knowledge, and extending the domain from $H_1(x)$ to $H_1(x) \cup D_S$ is a background defeater of "t is unreachable within one pixel".

Proof. (1) The one-pixel region of the reference is empty, so complete knowledge of it covers no state. (2) By E3, $[\![A \wedge [x_j = u_j]]\!] \subseteq R$ and $[\![A \wedge [x_j \geq c]]\!] \cap R = \emptyset$, so $u_j < c$ and every defeating state has $x_j \neq u_j$; on a single domain, antecedent strengthening is valid ($[\![A]\!] \cap D \subseteq R$ implies $[\![A \wedge C]\!] \cap D \subseteq R$). (3) $R^{(1)}_t(x) = \emptyset$ makes the statement true on $H_1(x)$, and any state of $R^x_{t,S}$ refutes it on the extension; such a state differs from x in both pixels of S, so it is outside every one-pixel condition. □

At θ_0, BASE holds for every $t \neq 3$, so $P_1(t;0,j)$ and $P_1^{\mathrm{core}}(t)$ are false for every t and j: introducing $P_1(t;0,j)$ means installing a one-pixel sufficient condition together with its strong defeater. On the pair $S = \{69, 557\}$ (anchor values 0), θ_0 itself satisfies $P_{\mathrm{hor}}(1;S)$, with 11,404 pair states of class 1 (we call this the *flagship* certificate), and fails $P_{\mathrm{hor}}(t;S)$ for every $t \in \{0,2,4,5,6,7,8,9\}$.

Attribution levels for training. A training-mediated introduction $\theta_1 = \mathrm{Train}(D \oplus_\pi z^*)$ is compared with its *matched control* $\theta_C = \mathrm{Train}(D)$: the same initialization, seed, data order, optimizer and schedule, without the crafted example. Let $\Delta = \theta_1 - \theta_C$.

Definition 3 (Attribution levels)**.** T-R1 *(relational introduction): $P_1(t;0,j)$ is false at θ_C and true at θ_1.* T-R2 *(component introduction): T-R1, and Δ is necessary for each component separately: restoring the control's first-layer column of pixel 0 in θ_1 removes $P_1^{\mathrm{core}}(t)$, while restoring the control's column of pixel j keeps $P_1^{\mathrm{core}}(t)$ with the same route region and removes E3(j).* T-R3 *(same law): a trained and a direct realization have the same law of class t on the declared domain $H_1(x) \cup D_{\{0,j\}}$.*

T-R2 excludes the case in which the blocking pixel's influence predates training and becomes a defeater only because a new route was installed: then restoring its column changes nothing. The ablation acts on one input column; it does not localize the block further inside the network.

III. Mechanism D: direct synthesis

Construction. A hidden unit is *dormant on* a set if its pre-activation under θ_0 is ≤ 0 on every element. Unit 25 is dormant on the 70,000 MNIST images, on the one-pixel horizon and on the pair states used below (maximum pre-activations -0.033, -1.71 and -1.83), but it is not dead: its pre-activation over $[0,1]^{784}$ is bounded only by +13.21.

Definition 4 (Dormant-unit edits)**.** *For a dormant unit k, pixels $i \neq i'$, amplitude $\omega > 0$, threshold τ_{255}, output weight $C_w > 0$ and class t, set $W_1[k,\cdot] = 0$ except $W_1[k,i] = \omega$ and $W_1[k,i'] = \mp\omega$; $b_1[k] = -\mathrm{fl}(\omega\tau)$ with $\tau = \mathrm{fl}(\tau_{255}\mathrm{fl}(1/255))$; and $W_2[\cdot,k] = 0$ except $W_2[t,k] = C_w$. With $-\omega$ (one-pixel edit $\mathcal{E}_1$), $h_k \approx \mathrm{ReLU}(\omega(x_i - x_{i'} - \tau))$; with $+\omega$ (two-pixel edit $\mathcal{E}_2$), $h_k \approx \mathrm{ReLU}(\omega(x_i + x_{i'} - \tau))$.*

The edit writes into an existing unit the ReLU indicator-and-AND pattern of REASSURE's support networks [4] and of hand-built backdoor gates [3]; it is not optimized and not claimed minimal.

Lemma 1 (Silence and inheritance)**.** *(i) Under $\mathcal{E}_1$, for every uint8 input with $u_i \leq \tau_{255}$ the new unit's pre-activation under $\mathcal{S}$ is ≤ 0. (ii) Under $\mathcal{E}_2$ with $\tau_{255} \geq 256$, it is negative for every input with $u_i + u_{i'} \leq 255$. (iii) At an input where the new unit is inactive under θ_1 and the original unit is inactive under θ_0, the two parameter sets compute the same logits up to the sign of a zero, hence the same class.*

The proofs, which track binary32 rounding, are in section A. Consequently, for $\mathcal{E}_1$ on pixels that are 0 in every reference image and in the anchor, no reference prediction changes, every one-pixel law of a pixel other than i is unchanged, and every new one-pixel target state changes pixel i; for $\mathcal{E}_2$ on pixels with anchor value 0, the whole one-pixel horizon is unchanged, so $R^{(1)}_t = \emptyset$ is inherited from θ_0. The analysis does not give the existence or shape of the target region, nor the defeater; those depend on θ_0's margins and are settled by exhaustive evaluation.

One-pixel result (exhaustively witnessed; table II, fig. 1). With rules fixed before execution (pixels i, j the two lowest-index pixels that are 0 in all reference images and the anchor, giving $(0,1)$; unit 25, $\omega = 10$, $\tau_{255} = 128$, C_w the first of $\{10, 30\}$ that works), $P_1(t;0,1)$ is introduced for 9/9 classes, all with $C_w = 10$. All 99–123 new one-pixel target states are on pixel 0; its region is an up-set $[a, 255]$ with a from 133 to 157; pixel 1 defeats it from $c = 99$ to 123. On the pair domain the target region is exactly the difference set $\{v_0 - v_1 \geq a\}$, so $a + c = 256$. Each edit changes 795 parameters, 0 of 70,000 MNIST predictions, and no one-pixel law of any other pixel.

Two-pixel result. On the support $S = \{69, 557\}$ (anchor values 0), where θ_0 certifies $k^* = 2$ for class 1 with 11,404 pair states (the *flagship* certificate), $\mathcal{E}_2$ ($\tau_{255} = 480$) introduces $P_{\mathrm{hor}}(t;S)$ for every class absent from the pair law, 8/8, with 28–210 pair states of class t, the one-pixel horizon unchanged, 0 reference changes and the flagship certificate preserved in every case (section F, table XV).

IV. Mechanism T: what one example does to the parameters

Write the network as $f(x) = W_2 h + b_2$, $h = \mathrm{ReLU}(z)$, $z = W_1 x + b_1$. For one example (x^*, y^*) with $x^* \in [0,1]^{784}$, cross-entropy gives the output error $e = \mathrm{softmax}(f(x^*)) - \mathbf{e}_{y^*}$ and the hidden error $\delta = (W_2^\top e) \odot \mathbf{1}[z(x^*) > 0]$, so

$$\nabla_{W_2} = e\, h^\top, \qquad \nabla_{b_2} = e, \qquad \nabla_{W_1} = \delta\, x^{*\top}, \qquad \nabla_{b_1} = \delta \tag{1}$$

[10]. In a batch the trainer averages over the batch.

TABLE II

DIRECT ONE-PIXEL INTRODUCTION (EDIT $\mathcal{E}_1$, PIXELS $i = 0$, $j = 1$), EXHAUSTIVE UNDER $\mathcal{S}$. R_0: ONE-PIXEL REGION OF PIXEL 0; $|R_t^{(1)}|$: TARGET STATES AMONG ALL 200,704 ONE-PIXEL STATES (ALL ON PIXEL 0); c: DEFEATER THRESHOLD; $|R^x_{t,\{0,1\}}|$: TARGET STATES OF THE PAIR LAW; CHANGES: PREDICTIONS CHANGED ON THE 70,000 MNIST IMAGES; LAW HASH: SHA-256 PREFIX OF THE ONE-PIXEL CLASS LAW.

t	R_0	$\lvert R_t^{(1)}\rvert$	c	$\lvert R^x_{t,\{0,1\}}\rvert$	E1	E2	E3	P_1	$\lVert\Delta\theta\rVert_2$	changes /70k	law hash
0	$[149, 255]$	107	107	5,778	true	true	true	true	18.03	0	`4cdcfed7`
1	$[133, 255]$	123	123	7,626	true	true	true	true	18.05	0	`73da9282`
2	$[145, 255]$	111	111	6,216	true	true	true	true	18.15	0	`eba65ad8`
4	$[157, 255]$	99	99	4,950	true	true	true	true	18.16	0	`304ca1b0`
5	$[141, 255]$	115	115	6,670	true	true	true	true	18.11	0	`171682e3`
6	$[150, 255]$	106	106	5,671	true	true	true	true	18.03	0	`e5fd391e`
7	$[152, 255]$	104	104	5,460	true	true	true	true	18.05	0	`9d5d294c`
8	$[142, 255]$	114	114	6,555	true	true	true	true	18.01	0	`78523e3c`
9	$[144, 255]$	112	112	6,328	true	true	true	true	18.22	0	`9944a938`

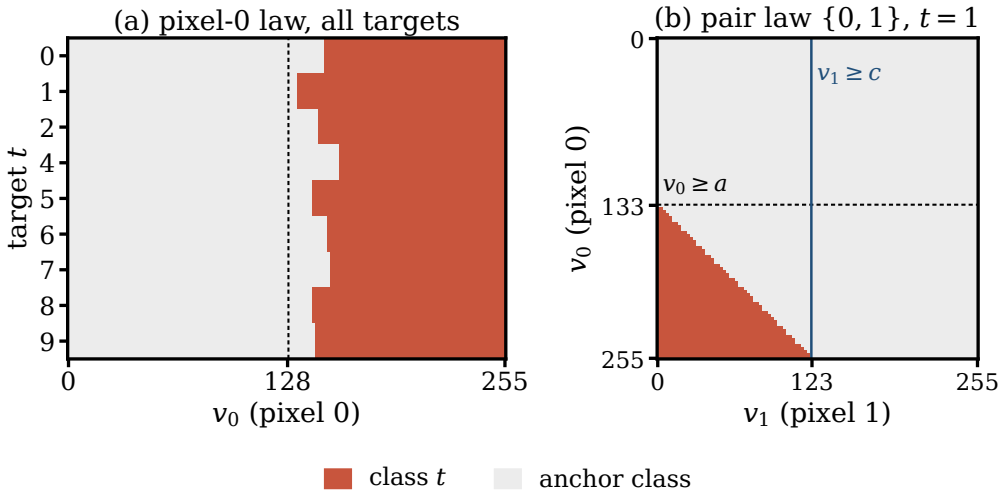

Fig. 1. Direct one-pixel edit. (a) Law of pixel 0, one row per class: the region is an up-set beginning above $v_0 = 128$, below which Lemma 1 guarantees silence. (b) Pair law on $\{0, 1\}$ for $t = 1$; the row $v_1 = 0$ realizes $A = [x_0 \geq a]$ and the block $v_1 \geq c$ defeats it.

Proposition 2 (Single-example updates)**.** (G1) *A unit inactive at x^* receives zero gradient from the example on its row of W_1, its bias and its column of W_2.* (G2) *Every lit pixel's incoming weight to a unit receives a contribution of the same sign.* (B1) *For $y^* = t$ and every unit k active at x^*, the example's descent direction raises $W_2[t, k]$ by $(1 - p_t)h_k \geq 0$ and lowers $W_2[c, k]$ by $p_c h_k \geq 0$ for $c \neq t$.* (G3) *Column k of W_1 receives gradient only from examples with $x_k > 0$; if pixel k is 0 in every image of the dataset D, its column is written only at the steps whose batch contains z^*, and it has no effect on any input with $x_k = 0$, in particular on every image of D and on the anchor.*

Proof. Immediate from (1): $\delta_k = h_k = 0$ for an inactive unit; $x^* \geq 0$; $e_t = p_t - 1 \leq 0$, $e_c = p_c \geq 0$ and $h_k \geq 0$; the batch gradient of column k is the average of $\delta^{(n)} x_k^{(n)}$. Under $\mathcal{S}$, a zero product leaves every fold unchanged (section A). □

In MNIST, 67 of the 784 pixels are 0 in every training image; pixels 0 and 1 are among them. Biases, W_2 and the columns of lit pixels are shared with the data, and Adam divides by a second moment accumulated from the whole dataset. The private column of a never-lit pixel is different, because Adam is invariant to the scale of the gradient [11, Sec. 2.1]. With $\rho = (1 - \beta_1)/\sqrt{1 - \beta_2}$, $r = \beta_1/\sqrt{\beta_2}$ and the trainer's $\beta_1 = 0.9$, $\beta_2 = 0.999$, let $\kappa = \rho/(1 - r) = 31.8$.

Lemma 2 (Exposure carry on a private coordinate)**.** *Let a scalar parameter be updated by Adam from zero moments with gradient non-zero only at n exposure steps. (i) Scaling every exposure gradient by $\gamma > 0$ changes each update only through the ϵ term. (ii) The total displacement is at most $n\kappa\,\mathrm{lr}$ for any gradient magnitudes and signs. (iii) One exposure at step s followed by zero gradients moves the parameter, for $\epsilon \to 0$, by an explicit sum that tends to $\kappa\,\mathrm{lr}$ and equals $3.01\,\mathrm{lr}$ at the last step of our runs.*

Proposition 3 (Coupling of private columns, G5)**.** *If pixels i, k are 0 in every image of D and $x_i^* > 0$, then under Adam from zero moments the updates of columns i and k have the same sign and differ only through ϵ; if $x_k^* = 0$, column k never changes.*

Proofs are in section A. These facts locate where a defeater of a trained antecedent can come from. Through a private column, the example cannot write one: a lit private pixel

becomes a second copy of the antecedent route (G5), and an unlit one keeps its initialization. Through the output layer, the example's own writes favour t (B1). At the baseline, no single pixel switches off any of the 82 units active at the anchor with pixel 0 raised (783 pixels checked in float32). What remains is the rest of the training run: the dataset's gradients in a trajectory that the example has changed. Section VII measures exactly that.

Design of the example. $x^* = I_x(\{0\}, 255)$, the anchor with the never-lit pixel 0 set to 255, labelled $y^* = t$. By G3 its private-column write acts only on states with $x_0 > 0$; by G1 only the 82 units active at x^* are written at θ_0; unit 25, which the direct edit rewrites, has pre-activation -1.80 at x^*, so the two mechanisms start from disjoint parameter supports. The same example is used at every schedule and for every class; only its label and its schedule change.

V. Matched training design

Learner and control. Train is the trainer that produced θ_0, verbatim: Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, learning rate 10^{-3}), batch 128, seed 0, the same per-epoch permutations, 5 epochs on the 60,000 MNIST training images, 2,345 Adam steps. In the default BLAS thread envelope the run without insertion reproduces θ_0 bit for bit (parameter SHA-256 prefix `62063c2c`), re-verified for this work. Hence $\theta_C = \theta_0$: the matched control of every treatment is the baseline itself. Runs with 1, 2, 3, 4 or 8 BLAS threads do not reproduce θ_0 and are not used for any certified model.

Schedules. The crafted example is appended to the scheduled batches (129 images): *T3* the last batch of the run (one exposure); *T4* the last batch of every epoch (5); *R1* every 50th batch (45); *Ek*, $k \in \{75, 100, 150\}$, every kth batch (30, 20 and 15 exposures). There is never more than one distinct crafted example; the exposure count belongs to the schedule.

Declarations. For P_1^{core}, the learner, envelope, domain, predicate, example, classes and schedules T3, T4 and the conditional relaxation R1 were sealed before any treatment run (preregistration SHA-256 prefix `780bc27df7ae`); R1 was triggered by the null result of T3 and T4. The defeater stages (a defeater search in the R1 models, then the E schedules for all classes) were declared afterwards in a written plan whose order is documented by file-system times. The attribution study was declared in a second plan before any of its computations, with four appended blocks, each appended before the computation it governs and after the results that motivated it: the sham specificity control and the certification rule; the T-R3 linear program; the exact form of its silence condition, after its first cases; and the exact test of near-feasible optima (section B). Every schedule–class cell and every attribution case is reported.

VI. Training-mediated synthesis

The antecedent alone. T3 and T4 install P_1^{core} for no class. R1 installs it for $t \in \{0, 4, 5, 6, 7, 9\}$ (6/9): each class is reached by changing pixel 0 to any value of an up-set $[a, 255]$, a from 102 to 176, and by no change of any other single pixel (fig. 4). Class 1 collapses (the anchor itself becomes class 1) and classes 2 and 8 leak to other pixels. The single-exposure displacement of the private column matches Lemma 2(iii) to 0.08% (section C). Training on the example alone, without the dataset, installs nothing in a screen and in a separately preregistered search (section D): alone, every lit pixel of the example is private to it.

Antecedent and defeater. Table IX reports every schedule–class cell. P_1^{core} holds for 6, 7, 8 and 3 of 9 classes under R1, E75, E100 and E150; class 1 is never reached. Sparser schedules write a weaker route (higher a): it stays confined to pixel 0 but can be defeated, while dense schedules write a strong route that leaks or collapses and out-competes every other pixel. Selecting per class the first schedule (in the order R1, E75, E100, E150) with an exact defeater and its lowest-index defeating pixel gives certified, bit-reproducible models for $t \in \{0, 2, 4, 5, 6, 7, 8, 9\}$ (8/9; table XI), with 15–45 exposures, test accuracy 97.20–97.35% (baseline 97.3%) and 32–45 test predictions changed. So $P_1(t; 0, j)$ is introduced by training (T-R1) for 8 classes. Unlike the direct edit, where pixel $j = 1$ is fixed in advance, the training route does not prescribe the blocking pixel: it is found by an exhaustive screen of all 783 candidates with exact confirmation, and the schedule by the declared selection rule. Whether training's change is necessary for the block is the next question.

VII. Exact attribution of the training effect

Why a separate attribution. T-R1 alone does not say that training produced the defeat: E3(j) is false at θ_C simply because the antecedent is absent there, and the defeating pixel's influence may predate training. The column ablations of Definition 3 separate the two components against the matched control. Because the matched control is θ_0, an ablation restores exactly baseline parameters.

Proposition 4 (Which pixels training can make defeaters)**.** *Let pixel j be 0 in every image of D and in x^*. Then column j of W_1 is identical in θ_1 and θ_C, so restoring it changes nothing, and no T-R2 defeat by pixel j exists. In particular, a never-lit pixel with anchor value 0, such as the direct edit's defeating pixel 1, can be a defeater of a trained antecedent only by inheritance.*

Proof. By G3 the gradient of column j is zero at every step of both runs, Adam's moments stay zero and so does every update; both runs start from the same initialization. □

So a T-R2 defeater must be a pixel the data light, whose column the dataset keeps writing during the run. Every certified defeater of table XI has anchor value 0, and x^* differs from the anchor only at pixel 0, so the example's own gradient on the defeating column is zero at every exposure: any training-written defeat is written by the dataset's gradients in the trajectory the example perturbed.

Sweep. We declared and ran a two-stage search over the saved treatment models (section B): a float32 screen of E3(j) for all 783 pixels j, a float32 screen of the column

ablation for every screened candidate, and exact pair laws for every candidate whose ablation screen removes or weakens the defeat, until the first exact T-R2 per model. For class 0, before the stopping rule took effect, the exact stage covered all 76 screened candidates (of 104) whose screened ablation changed the defeat: 8 pixels are exact T-R2 defeaters; for 66 the ablation keeps the defeat at a higher threshold, i.e. training strengthened a defeat that already acted; for 1 training weakened it, and for 1 the ablation also breaks the antecedent. The remaining 28 are unchanged in the screen. The direction is not symmetric: training overwhelmingly strengthens the defeaters of the antecedent it installed.

Certified attribution. For every class with a T-R2 hit we retrained the selected model three times from scratch and evaluated control, treatment and both ablations exhaustively (table III): T-R2 holds for $t \in \{0, 2, 4, 5, 6, 7, 8, 9\}$ (8/9), every treatment rerun is bit-identical, the matched control is θ_0 bit for bit, restoring the antecedent column removes P_1^{core} in every case, and restoring the defeating column keeps the antecedent region and removes the defeat. The defeating pixels are lit in 375–41,786 of the 70,000 MNIST images and 0 in the anchor; defeater thresholds range from 165 to 249, antecedents from 154 to 249, with 15–30 exposures, test accuracy 97.20–97.33% and 27–45 test predictions changed. The highest defeater thresholds (248 and 249) lie close to the top of the pixel range; the certification is enumerative over all 256 values, so this does not weaken it, but it means those defeats act only at nearly saturated values of pixel j. For classes 2, 5 and 7 the T-R2 model comes from a different schedule than the first-found defeater model of table XI; in that model the checked defeaters were only strengthened by training (classes 2 and 7) or left unchanged (class 5). The selection rule is stated in section B.

Specificity. Inserting any example perturbs the whole trajectory: the treatment changes almost every parameter slightly (section C). A defeat that disappears when the control column is restored could therefore be a generic consequence of perturbation. A sham run separates the two: the same crafted input x^*, the same schedule, labelled with the anchor's own class 3, so that it installs no route to t. Substituting the sham run's column of pixel j into the treatment removes the defeat for 6 of the 8 certified classes (one model and one defeating pixel per class, table III), so there the defeat depends on the column the target-labelled run wrote, not on perturbation as such; in the remaining 2 (classes {0,6}) the sham column weakens the defeat without removing it (part of the effect is generic). The sham test was also run on every other T-R2 pixel found by the sweep, which are not independent of the certified rows: the seven further class-0 pixels and a second class-5 model. Over these 16 (model, pixel) pairs it removes the defeat for 8; for class 0 alone it does so for one of eight pixels, so class 0's defeats are mostly a generic effect of perturbation strengthened by the target-labelled run.

Mechanism. The account that fits these facts, and that we state as a conjecture with this evidence: the example raises t's output weights on the units it activates (B1); dataset images that share those units now lean towards t, and their gradients push back by strengthening, in the shared columns of the pixels they light, influence against t. A pixel lit by images of a competing class thus acquires influence against t that the private antecedent pixel, never lit by the data, does not share, and at high values it blocks the antecedent. The defeat is written by the dataset's corrective response to the example, a second-order effect that the first-order analysis of section IV leaves open; Proposition 4 shows that it is the only channel for an unlit pixel.

VIII. Direct versus trained realization

Figure 2 and table IV set the two mechanisms side by side on one target. On each certified pair $\{0, j\}$ the direct edit (with pixel j in place of pixel 1) also satisfies $P_1(t; 0, j)$ exactly, with no MNIST prediction changed: on all 8 attribution pairs (antecedents from 141 to 157, defeater thresholds from 99 to 115) and on the first-found pairs (table XII). The parameter paths differ. The direct edit changes 795 parameters of unit 25, which is inactive at x^*, so the example's own gradient has no component on them; the full training change moves almost every parameter by a small amount (section C). The direct edit writes the defeat into the edited unit ($-\omega$ from pixel j); training writes it through the dataset's response.

IX. Same law along different parameter paths

Agreement on a predicate is weaker than agreement on the law: two networks can satisfy $P_1(t; 0, j)$ with different thresholds and regions. We now ask whether the two mechanisms can produce the *identical* finite law, and what their parameter changes look like when they do. This separates two levels that are easy to conflate: equality of the exact input–output law on a declared intervention domain, and similarity of the parameter changes that realize it.

With its fixed constants the direct edit reproduces the predicate, not the law: on every certified pair its antecedent and defeater thresholds differ from the trained ones (table XII). We therefore searched the direct family itself, unit 25 with free input weights w_0, w_j, bias b and output weight C_w to t only, for a member with the trained law. Let $m_0(s) > 0$ be the margin of θ_0 without unit 25 against t at a state s. The edited network classifies s as t if and only if $C_w \,\mathrm{ReLU}(w_0 x_0 + w_j x_j + b) > m_0(s)$, and since $m_0(s) > 0$ on every state (verified exhaustively for every case; the minimum over all eight pairs is 4.80) this is the linear condition $C_w(w_0 x_0 + w_j x_j + b) > m_0(s)$. Requiring the trained law's label on all 65,536 pair states, and silence on the one-pixel states of every other pixel, is a linear program in $(C_w w_0, C_w w_j, C_w b)$; we maximize its margin in real arithmetic, with the scaled weights boxed in $[-10^4, 10^4]$, and accept only the exact binary32 law of the edit built from the solution. In both accepted cases $C_w w_0$ sits at the box bound ($w_0 = 1000$ with $C_w = 10$): the bound is a declared constant, and a different one gives a different, equally exact edit or none. The linear program is a standard single-layer repair tool in the sense of Sotoudeh and Thakur [12]; what is new is its target, the complete law of a trained model.

TABLE III

TRAINING ATTRIBUTION CERTIFICATES (EXHAUSTIVE UNDER $\mathcal{S}$; MATCHED CONTROL $= \theta_0$ BIT FOR BIT). SCHEDULE AND EXPOSURES OF THE ONE DISTINCT EXAMPLE; ANTECEDENT REGION AND DEFEATER (j, c) OF THE TREATMENT; "$-$COL 0": P_1^{core} AFTER RESTORING THE CONTROL COLUMN OF PIXEL 0; "$-$COL j": ANTECEDENT KEPT / DEFEAT AFTER RESTORING THE CONTROL COLUMN OF PIXEL j; SHAM: E3(j) AFTER SUBSTITUTING THE SHAM-TRAINED COLUMN (SECTION VII); RERUNS: THREE FRESH RUNS BIT-IDENTICAL; MNIST TEST ACCURACY AND TEST PREDICTIONS CHANGED.

t	schedule	exp.	region	(j, c)	lit	$-$col 0	$-$col j	sham	reruns	test % / changes
0	E100	20	$[220, 255]$	$(268, 227)$	41,786	P_1^{core} lost	kept / removed	weakened (c 247)	3/3	97.33 / 35
2	E150	15	$[213, 255]$	$(100, 244)$	5,880	P_1^{core} lost	kept / removed	defeat removed	3/3	97.25 / 33
4	E100	20	$[249, 255]$	$(525, 165)$	16,221	P_1^{core} lost	kept / removed	defeat removed	3/3	97.20 / 45
5	E100	20	$[242, 255]$	$(655, 193)$	34,810	P_1^{core} lost	kept / removed	defeat removed	3/3	97.24 / 38
6	E75	30	$[195, 255]$	$(145, 240)$	969	P_1^{core} lost	kept / removed	weakened (c 245)	3/3	97.25 / 38
7	E100	20	$[215, 255]$	$(121, 241)$	4,430	P_1^{core} lost	kept / removed	defeat removed	3/3	97.26 / 27
8	E150	15	$[154, 255]$	$(676, 249)$	375	P_1^{core} lost	kept / removed	defeat removed	3/3	97.29 / 34
9	E150	15	$[200, 255]$	$(655, 248)$	34,810	P_1^{core} lost	kept / removed	defeat removed	3/3	97.26 / 32

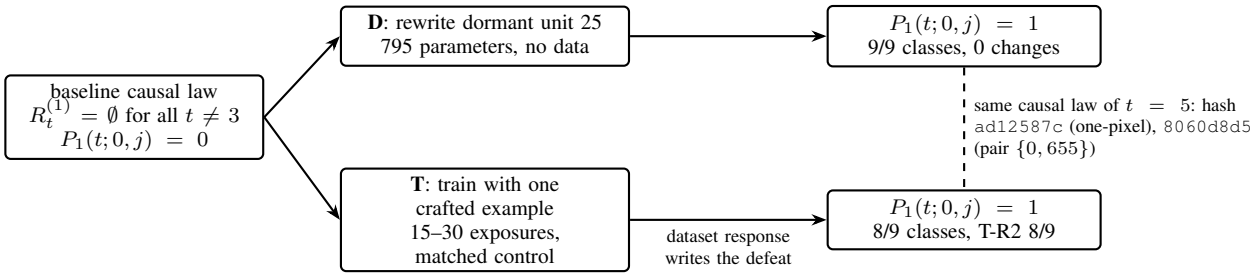

Fig. 2. One target, two mechanisms. From the baseline causal law (no one-pixel route to t), a direct edit of one dormant unit and a training run with one crafted example both reach a law satisfying $P_1(t; 0, j)$, verified exhaustively under $\mathcal{S}$. Numbers are generated from the certificates.

TABLE IV

DIRECT AND TRAINING-MEDIATED SYNTHESIS OF THE SAME PREDICATE. VALUES ARE RANGES OVER THE CERTIFIED CLASSES; ALL LAW PROPERTIES ARE EXHAUSTIVE UNDER $\mathcal{S}$.

	mechanism D (direct edit)	mechanism T (training, one example)
baseline	θ_0, causal law with $R_t^{(1)} = \emptyset$	θ_0 (= matched control, bit for bit)
target predicate	$P_1(t; 0, j)$ (Definition 1)	$P_1(t; 0, j)$ (Definition 1)
classes introduced	9/9 with $j = 1$; 8/8 on the trained pairs	8/9 (T-R1); class 1 never
attribution	by construction	T-R2 for 8/9; target-specific defeat 6/8
same causal law (T-R3)	2 classes ($t \in \{5, 7\}$), LP-fitted unit	2 classes ($t \in \{5, 7\}$)
parameters changed	795 (one unit)	91,563–91,564 (T-R3 models)
crafted examples / exposures	none	1 distinct / 15–30
MNIST predictions changed	0 of 70,000	27–45 of 10,000 test
test accuracy	97.3% (unchanged)	97.20–97.33%
defeat written by	the edit ($-A$ from pixel j)	the dataset's response to the example
acceptance	exhaustive, binary32	exhaustive, binary32, 3 bit-identical reruns

Result (T-R3). For $t = 5$ (pixel 655, LP margin 0.151) and $t = 7$ (pixel 121, margin 0.126), the edit built from the solution has exactly the trained law of class t on the declared domain $H_1(x) \cup D_{\{0,j\}}$: the target states of all 200,704 one-pixel interventions and all 65,536 pair interventions coincide (table V). For $t = 5$ the agreement extends to the full class law on both domains; for $t = 7$, 358 one-pixel states receive different non-target classes, which the law of class t does not record. Both laws carry the same route and block thresholds, so the same sufficient condition and the same defeater are realized twice. The realizations are reached along very different parameter changes (fig. 3): the edit changes 795 parameters of one unit, training changes 91,563 (for $t = 5$), the two supports share 212 coordinates, and the cosine between the two changes is -7.6×10^{-7}. The trained realization of the same law also carries the T-R2 attribution of section VII. The two models are not the same function: the edit changes no MNIST prediction, the trained model changes a few dozen (table III).

Interpretation. That different parameters can compute the same function is known for ReLU networks [13], equivalence of two given networks can be verified [14, 15], and many models fit the same data equally well [16]. What the measurement adds is that two different synthesis mechanisms, an explicit edit and a training run, reach the identical exact law on a declared intervention domain while their parameter changes differ in size and support by orders of magnitude (a cosine near zero is expected for a 795-sparse and a dense change, so we do not read it as evidence on its own). Equality of the law therefore does not constrain the parameter change here. We do not infer any equivalence of internal representations, and the statement concerns the declared domain only (section X).

The other classes. For the other 6 classes the best LP margin is negative (from -0.0357 to -0.0004): in real

TABLE V

SAME LAW, DIFFERENT PARAMETER PATHS (T-R3). FOR EACH CLASS, THE LP-FITTED ONE-UNIT EDIT AND THE TRAINED MODEL (TABLE III) ARE EVALUATED EXHAUSTIVELY UNDER $\mathcal{S}$ ON ALL ONE-PIXEL AND PAIR INTERVENTIONS. LAW HASHES: SHA-256 PREFIXES OF THE LAW OF CLASS t; THEY ARE IDENTICAL FOR THE TWO MECHANISMS. PARAMETER COLUMNS: COORDINATES CHANGED RELATIVE TO θ_0, SHARED COORDINATES, COSINE AND EUCLIDEAN NORM OF THE TWO CHANGES.

t	j	LP margin	edit $(w_0, w_j, b; C_w)$	route / block	law hash H_1	law hash pair	class law on H_1	params D / T	shared	cosine	$\|\Delta\|_2$ D / T	test changes D / T
5	655	0.151	(1000, −69.21, −947.31; 10)	[242, 255] / 193	`ad12587c`	`8060d8d5`	identical	795 / 91,563	212	-7.6×10^{-7}	1379 / 3.02	0 / 38
7	121	0.126	(1000, −167.70, −841.20; 10)	[215, 255] / 241	`00643aab`	`76cf5dca`	358 differ	795 / 91,564	211	-1.4×10^{-6}	1317 / 2.82	0 / 27

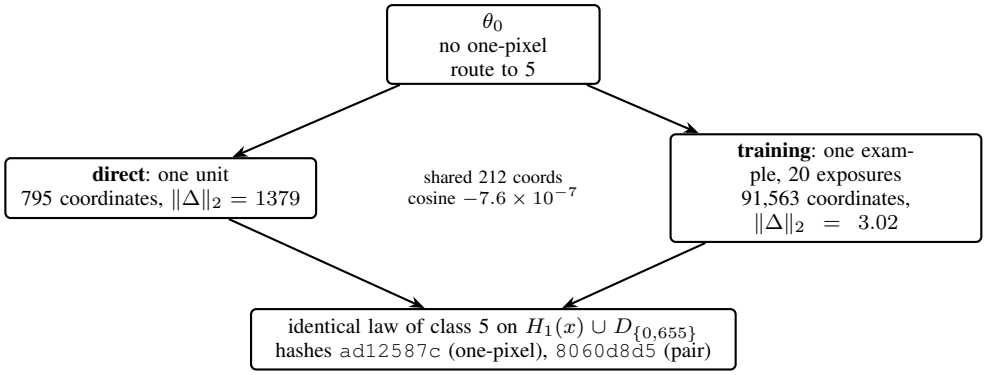

Fig. 3. Two parameter paths, one law ($t = 5$). Left: the direct edit concentrates a large change on one unit; right: training spreads a small change over almost all parameters. Both end in the identical law of class 5 on the declared domain (law hashes shown). Numbers are generated from the T-R3 record.

arithmetic every member of the one-unit family with scaled weights in $[-10^4, 10^4]$ fails the trained law at some state of the declared domain, also when the silence condition is relaxed to its exact form (0 further classes), and the edits built from the near-feasible optima for $t \in \{4, 6, 8\}$ mislabel 5–10 pair states under binary32. Their trained laws are not half-planes in θ_0's margin geometry with scaled coefficients in $[-10^4, 10^4]$; we did not bound the gap between the real and the binary32 law, so for them T-R3 is not reached within this family, which is not a statement about richer edits.

X. SAME LAW, DIFFERENT EXTENSION

The predicates read interventions outside the pair domain: P_{hor} through $R_t^{(1)}$ and P_1 through E1. Write $\Lambda^D(f)$ for the class law of f on D, and say a property *factors through* Λ^D if it is a function of it.

Lemma 3 (Restricted equivalence, elementary)**.** *Let $D = D_S$ with $S = \{p, q\}$. (1) If $\Lambda^D(f) = \Lambda^D(f')$, every property that factors through Λ^D agrees: the region and its model count, the Booleanized law and its ROBDD, every sufficiency and defeater query on D, and E2 and E3 for $(0, j) = (p, q)$. (2) $P_{\mathrm{hor}}(t; S)$ and E1 do not factor through Λ^D: for every class law L on D admitting them there are classifiers with law L on D for which they hold and fail.*

Proof. (1) Equal arguments give equal values. (2) $H_1(x) \setminus D \neq \emptyset$: change f at one image $x[r := v]$ with $r \notin S$ to class t; the law on D is unchanged and $R_t^{(1)}$ gains a state at pixel r. □

In table VI, $P_{\mathrm{loc}}(t; S)$ is the pair-local form of P_{hor}: no target state on the two one-pixel slices through S inside D, and some target state in D; it factors through Λ^D. A specification or acceptance test stated on D alone (an exact repair specification, a law hash, an ROBDD) therefore certifies E2 and E3 but not P_{hor} or E1. The lemma is about functions; inside one trained architecture it is realized too. On a deeper 784–128–128–128–10 MLP, three exact realizations of one two-pixel law edit (indicators planted at different depths, routing solved by a linear program, no gradient training) have the identical law hash on all 65,536 states, so every property that factors through the law agrees, yet the deepest sends 264 one-pixel states outside D to the target and loses P_{hor} (table VI).

XI. RELATED WORK

Table VII places the closest work against the three ingredients of this paper: the kind of target, the mechanism, and the guarantee. Each has precedents; we did not find work that combines them.

One-example and one-poison results. A single clean-label poison changes one target prediction in transfer learning [5], and a single optimized image backdoors deep networks [6]. One poisoner can steer a linear model [31]; one poison provably backdoors regularized linear models and, with a random poison direction, wide bias-free k-layer ReLU networks under gradient flow, and a direction unused by the benign data, such as pixels constant in the training set, is singled out as the favourable case [7]; the guarantee for unused directions is proved for linear models. Our routing pixel is such a direction, and our antecedent result is a witnessed instance of the same mechanism class in a setting outside those theorems' hypotheses (biases, width 128, Adam, a repeated example). We differ in the target (a predicate on a complete finite law, with a no-other-pixel clause and a defeater), the guarantee

TABLE VI

EXTENSION ATLAS ON THE DEEPER MLP. THE LEFT BLOCK FACTORS THROUGH THE PAIR LAW (IDENTICAL HASH); THE RIGHT BLOCK NEEDS $H_1 \setminus D$ (200,704 ONE-PIXEL STATES, EXHAUSTIVE). $P_{\mathrm{loc}}(1)$: NO CLASS-1 STATE ON THE TWO ONE-PIXEL SLICES THROUGH S INSIDE D, AND SOME CLASS-1 STATE IN D.

	determined by Λ^D			needs $H_1 \setminus D$	
model	law hash on D	$\lvert R^x_{1,S}\rvert$	$P_{\mathrm{loc}}(1)$	$\lvert R^{(1)}_1\rvert$	$P_{\mathrm{hor}}(1)$
D0 (unedited)	`13ea1c53`	0	false	0	false
early-r	`a1ced8c7`	2,601	true	0	true
middle	`a1ced8c7`	2,601	true	0	true
late	`a1ced8c7`	2,601	true	264	false

TABLE VII

CLOSEST WORK. "TARGET": WHAT THE METHOD IS ASKED TO PRODUCE OR CHECK. "GUARANTEE": THE ARITHMETIC AND FORM OF THE ACCEPTANCE AS STATED IN THE SOURCE.

work	target	mechanism	guarantee	relation to this paper
Peinemann et al. [7]	trigger-to-label backdoor	one poison, training	probabilistic, gradient flow, wide nets	owns one poison and unused directions
Shafahi et al. [5]; Geiping et al. [17]; Souri et al. [18]	one target prediction or trigger	crafted poisons, training	empirical	target is a prediction, not a law
Hong et al. [3]	backdoor	direct edit of low-impact units	empirical	owns dormant-unit takeover
Zhu et al. [19]; Wang et al. [20]; Cazenavette et al. [21]	learner reaches a target model or accuracy	designed or distilled training data	varies; empirical for distillation	owns teaching towards a target
Fu and Li [4]; Sotoudeh and Thakur [12]; Tao et al. [22]	output constraints on points or regions	patch networks; LP repair of a layer	sound over the reals	own the indicator-and-AND pattern and LP layer repair
Fu et al. [23]; Chi et al. [24]	global robustness, fairness; local robustness	weight repair; patch modules	over the reals	repair a property, not synthesize a law
Shapira et al. [25]; Shapira et al. [26]; Katz et al. [27]; Tjeng et al. [28]	few-pixel or norm-bounded robustness	verification of a fixed net	complete over the reals	check a given network
Zhang et al. [29]; Narodytska et al. [30]	input–output behaviour of binarized nets	BDD / SAT compilation of the network	exact	compile the network; here only the finite law of a float32 net is Boolean
this paper	a predicate on the complete one-pixel and pair law at one image	one-unit edit; training with one distinct example	exhaustive, binary32; matched control	

(deterministic, per model, by enumeration under binary32), the design (a bit-identical matched control and a declared schedule; one distinct example, repeated), and the attribution of each component of the relation. Poisons acting through many examples [17, 18, 32] target a trigger or a prediction.

Direct edits, dormant units and repair. Handcrafted backdoors take over low-impact neurons and build gates from weights [3]; fine-pruning removes neurons dormant on clean inputs [33]; REASSURE's support networks own the indicator-and-AND construction [4]. Provable and architecture-preserving repair [12, 22–24, 34], verification-guided repair [35, 36] and retrain-and-certify methods [37] change a network until a point, region, robustness or fairness specification holds, with guarantees over the reals. The restriction of our target to the pair domain could be encoded as point specifications for such tools; we ran no such baseline and make no claim about what they can do on it. Direct editing of factual associations in language models [38, 39] shares the idea of an edit with measured locality; its object and acceptance differ.

Influence and teaching. Influence functions, TracIn and datamodels estimate the effect of one example or a data subset [40–42]; our matched control is the exact two-point counterfactual of one run, and the quantity is a law relation of the endpoint. Machine teaching asks for training data that drive a learner to a target, possibly a property of the learned model [19, 43], with training-set attacks as an instance [44]; iterative teaching and dataset distillation choose or synthesize few examples for a gradient learner [20, 45, 46]. The framing of a teacher with a target is therefore known; our target class is a predicate on an exact finite law, and our example is derived from the update analysis, not optimized.

Sparse attacks and their verification. One-pixel and sparse attacks find few-pixel label changes of a fixed network [47, 48]; complete verifiers certify few-pixel robustness [25, 26]. They examine a given network; here a prescribed one-pixel structure is installed and then checked exhaustively on one anchor. Guarantees over the reals need not hold for the deployed floating-point computation [49]; our acceptance is stated for the declared binary32 computation.

Position. To our knowledge, among the sources we inspected, no prior work takes a predicate on the complete finite intervention law of a float32 network (a unique one-pixel antecedent with a blocking second pixel) as a synthesis target, realizes that same predicate both by a closed-form edit of one dormant unit and by ordinary training with one distinct crafted example, and certifies each realization and the training attribution by exhaustive binary32 enumeration against a bit-identical matched control. The contribution is this combination

on a declared instance.

XII. Limitations

Scope. One baseline network (a small MLP), one dataset (MNIST), one anchor, one antecedent pixel, one optimizer configuration and seed, 9 classes and a small declared schedule family; the same-law witness uses one deeper MLP. Every positive statement is an exhaustive, deterministic fact about a saved model; none is a rate over anchors, pixels, seeds or networks. Determinism holds in the declared thread envelope only.

Channels. Both mechanisms use a pixel that is 0 in every MNIST image: the direct edit needs it for its locality lemma, and training needs it as a private channel that Adam writes at full size. Data without such features support neither mechanism in this form. The same property makes never-lit inputs a write channel for a single repeated example; freezing or monitoring input columns the data never activate would close it, and would also block the direct edit.

Attribution. T-R2 is a necessity statement at the granularity of one input column against the matched control; it does not localize the defeat inside the network, and the defeat also uses parameters that training did not change. The mechanism of section VII is a conjecture supported by the sham control and the direction asymmetry, not a proof. The searches are float32 screens with exact confirmation; a model without a confirmed candidate may still contain one, while every reported result is exact. The attribution plan was declared after the defeater stages, and the per-class selection of attribution models is stated with its order of declaration (section B).

Interpretation. "Paradox" names the interpretation of the defined relations, not an inconsistency of the network. Read as attacks, both mechanisms plant certified one-pixel targeted behaviour that reference-set evaluation would not detect; we report them on a small MNIST model as exact objects of study.

XIII. Conclusion

A formally specified sufficient-condition-with-defeater structure of an exact finite intervention law can be synthesized by two different mechanisms and certified exhaustively for both. A closed-form edit of one dormant unit writes the structure for every class with no prediction changed. One distinct crafted example, inserted into the model's own training run, also produces it: the example installs the antecedent through a private input channel, and the dataset's response to the example writes the defeat into the input weights of a pixel the data light, which a column ablation against a bit-identical matched control certifies for 8 of 9 classes. For two classes the two mechanisms meet exactly: a one-unit edit and the training run produce the same law of the target class on the whole declared domain, through parameter changes of very different size and support. Because these predicates read interventions outside the pair domain, exact agreement on a pair law does not fix them. Open questions are the class that training never reaches, anchors and data without private channels, and a proof of the corrective-response mechanism.

Reproducibility. The artifact contains the trainer and insertion drivers, the crafted example, every saved model, the exact law checker, the preregistration and plans with their hashes and file-system times, every certificate, the attribution and specificity records, and the generator of every number, table and figure, which recomputes or cross-checks each value from the frozen records and aborts on a mismatch. One command re-verifies the headline certificates from the saved models. The artifact is provided as anonymized supplementary material for review.

References

[1] J. L. Mackie, "Causes and conditions," *American Philosophical Quarterly*, vol. 2, no. 4, pp. 245–264, 1965. [Online]. Available: https://www.jstor.org/stable/20009173

[2] J. L. Pollock, "Defeasible reasoning," *Cognitive Science*, vol. 11, no. 4, pp. 481–518, 1987.

[3] S. Hong, N. Carlini, and A. Kurakin, "Handcrafted backdoors in deep neural networks," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 8068–8080.

[4] F. Fu and W. Li, "Sound and complete neural network repair with minimality and locality guarantees," in *International Conference on Learning Representations (ICLR)*, 2022.

[5] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[6] J. Hayase and S. Oh, "Few-shot backdoor attacks via neural tangent kernels," in *International Conference on Learning Representations*, 2023.

[7] T. Peinemann, P. Arnold, S. Berndt, T. Eisenbarth, and E. Mohammadi, "Provable one-poison backdoor attacks on linear models and ReLU neural networks," 2026, arXiv:2508.05600v3 [cs.LG], 27 Aug 2026; v1 (7 Aug 2025) had a different title.

[8] A. Geiger, D. Ibeling, A. Zur, M. Chaudhary, S. Chauhan, J. Huang, A. Arora, Z. Wu, N. Goodman, C. Potts, and T. Icard, "Causal abstraction: A theoretical foundation for mechanistic interpretability," *Journal of Machine Learning Research*, vol. 26, no. 83, pp. 1–64, 2025. [Online]. Available: http://jmlr.org/papers/v26/23-0058.html

[9] R. E. Bryant, "Graph-based algorithms for Boolean function manipulation," *IEEE Transactions on Computers*, vol. C-35, no. 8, pp. 677–691, 1986.

[10] I. Goodfellow, "Efficient per-example gradient computations," 2015.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR)*, 2015.

[12] M. Sotoudeh and A. V. Thakur, "Provable repair of deep neural networks," in *Proceedings of the 42nd ACM*

SIGPLAN International Conference on Programming Language Design and Implementation (PLDI). ACM, 2021, pp. 588–603.

[13] M. Phuong and C. H. Lampert, "Functional vs. parametric equivalence of ReLU networks," in *International Conference on Learning Representations (ICLR)*, 2020.

[14] S. Teuber, M. Kleine Büning, P. Kern, and C. Sinz, "Geometric path enumeration for equivalence verification of neural networks," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2021.

[15] C. Eleftheriadis, N. Kekatos, P. Katsaros, and S. Tripakis, "On neural network equivalence checking using SMT solvers," 2022.

[16] L. Semenova, C. Rudin, and R. Parr, "On the existence of simpler machine learning models," in *2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. ACM, 2022, pp. 1827–1858.

[17] J. Geiping, L. Fowl, W. R. Huang, W. Czaja, G. Taylor, M. Moeller, and T. Goldstein, "Witches' brew: Industrial scale data poisoning via gradient matching," in *International Conference on Learning Representations (ICLR)*, 2021.

[18] H. Souri, L. Fowl, R. Chellappa, M. Goldblum, and T. Goldstein, "Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 19 165–19 178.

[19] X. Zhu, A. Singla, S. Zilles, and A. N. Rafferty, "An overview of machine teaching," 2018.

[20] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," 2018.

[21] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, and J.-Y. Zhu, "Dataset distillation by matching training trajectories," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 708–10 717.

[22] Z. Tao, S. Nawas, J. Mitchell, and A. V. Thakur, "Architecture-preserving provable repair of deep neural networks," *Proceedings of the ACM on Programming Languages*, vol. 7, no. PLDI, pp. 443–467, 2023.

[23] F. Fu, Z. Wang, W. Zhou, Y. Wang, J. Fan, C. Huang, Q. Zhu, X. Chen, and W. Li, "REGLO: Provable neural network repair for global robustness properties," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 061–12 071.

[24] Z. Chi, J. Ma, P. Yang, C.-C. Huang, R. Li, J. Wang, X. Huang, and L. Zhang, "Patch synthesis for property repair of deep neural networks," in *IEEE/ACM International Conference on Software Engineering (ICSE)*, 2025.

[25] Y. Shapira, E. Avneri, and D. Drachsler-Cohen, "Deep learning robustness verification for few-pixel attacks," *Proceedings of the ACM on Programming Languages*, vol. 7, no. OOPSLA1, pp. 434–461, 2023.

[26] Y. Shapira, N. Wiesel, S. Shabelman, and D. Drachsler-Cohen, "Boosting few-pixel robustness verification via covering verification designs," in *Computer Aided Verification (CAV 2024)*, ser. Lecture Notes in Computer Science, 2024, pp. 377–400.

[27] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Computer Aided Verification (CAV 2017)*, ser. LNCS. Springer, 2017, pp. 97–117.

[28] V. Tjeng, K. Xiao, and R. Tedrake, "Evaluating robustness of neural networks with mixed integer programming," in *International Conference on Learning Representations (ICLR)*, 2019, arXiv:1711.07356.

[29] Y. Zhang, Z. Zhao, G. Chen, F. Song, and T. Chen, "BDD4BNN: A BDD-based quantitative analysis framework for binarized neural networks," in *Computer Aided Verification (CAV 2021)*, ser. LNCS. Springer, 2021, pp. 175–200.

[30] N. Narodytska, S. Kasiviswanathan, L. Ryzhyk, M. Sagiv, and T. Walsh, "Verifying properties of binarized deep neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018, pp. 6615–6624.

[31] L.-N. Hoang, "The poison of dimensionality," 2024.

[32] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47 230–47 244, 2019.

[33] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Research in Attacks, Intrusions, and Defenses (RAID 2018)*, ser. Lecture Notes in Computer Science, 2018, pp. 273–294.

[34] J. Ma, J. Wang, Q. Xuan, and Z. Wang, "Provable repair of deep neural network defects by preimage synthesis and property refinement," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2025, pp. 4169–4183.

[35] J. Ma, P. Yang, J. Wang, Y. Sun, C.-C. Huang, and Z. Wang, "VeRe: Verification guided synthesis for repairing deep neural networks," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering (ICSE)*, 2024.

[36] A. Fayyazi, Y. Xiao, P. Nuzzo, and M. Pedram, "Efficient counterexample-guided fairness verification and repair of neural networks using satisfiability modulo convex programming," in *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, 2025, pp. 367–375.

[37] F. Bauer-Marquart, D. Boetius, S. Leue, and C. Schilling, "SpecRepair: Counter-example guided safety repair of deep neural networks," in *Model Checking Software (SPIN 2022)*, ser. Lecture Notes in Computer Science, 2022, pp. 79–96.

[38] K. Meng, D. Bau, A. Andonian, and Y. Belinkov, "Locating and editing factual associations in GPT," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[39] K. Meng, A. Sen Sharma, A. Andonian, Y. Belinkov, and D. Bau, "Mass-editing memory in a transformer," in *International Conference on Learning Representations (ICLR)*, 2023.

[40] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, 2017, pp. 1885–1894.

[41] G. Pruthi, F. Liu, S. Kale, and M. Sundararajan, "Estimating training data influence by tracing gradient descent," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[42] A. Ilyas, S. M. Park, L. Engstrom, G. Leclerc, and A. Madry, "Datamodels: Understanding predictions with data and data with predictions," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 162, 2022, pp. 9525–9587.

[43] X. Zhu, "Machine teaching: An inverse problem to machine learning and an approach toward optimal education," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.

[44] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, 2015.

[45] W. Liu, B. Dai, A. Humayun, C. Tay, C. Yu, L. B. Smith, J. M. Rehg, and L. Song, "Iterative machine teaching," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, ser. PMLR, vol. 70, 2017, pp. 2149–2158.

[46] Z. Qiu, W. Liu, T. Z. Xiao, Z. Liu, U. Bhatt, Y. Luo, A. Weller, and B. Schölkopf, "Iterative teaching by data hallucination," in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, ser. PMLR, vol. 206, 2023, pp. 9892–9913.

[47] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.

[48] F. Croce and M. Hein, "Sparse and imperceivable adversarial attacks," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4723–4731.

[49] K. Jia and M. Rinard, "Exploiting verified neural networks via floating point numerical error," in *Static Analysis (SAS)*, ser. Lecture Notes in Computer Science, 2021, pp. 191–205.

APPENDIX A
PROOFS

Zero products in a binary32 fold. If a weight is finite and its input is a zero, the product is a signed zero; adding a signed zero to a nonzero accumulator returns it, and adding it to a zero gives a zero. Every claim below that a term "does not change" a pre-activation uses this, and assumes, as the evaluator asserts, that no NaN or infinity arises. This gives the second part of G3 (Proposition 2).

Proof of Lemma 1. (i) Rounding is monotone, so $u_i \le \tau_{255}$ gives $x_i = \mathrm{fl}(u_i \mathrm{fl}(1/255)) \le \tau$ and $p := \mathrm{fl}(\omega x_i) \le q := \mathrm{fl}(\omega\tau)$. The product $\mathrm{fl}(-\omega x_{i'})$ is ≤ 0 since $x_{i'} \ge 0$, and every other product is a zero. The fold therefore yields $\sigma = \mathrm{fl}(p + \mathrm{fl}(-\omega x_{i'})) \le p$, up to the sign of a zero, and adding the bias $-q$ gives $\mathrm{fl}(\sigma - q) \le 0$, so ReLU returns 0. (ii) Each x_k exceeds $u_k/255$ by a relative error below 2^{-23}, so $u_i + u_{i'} \le 255$ gives $x_i + x_{i'} \le 1 + 2^{-22}$. The pre-activation is the rounded value of $\omega x_i + \omega x_{i'} - \mathrm{fl}(\omega\tau)$, computed in at most four binary32 operations of relative error at most 2^{-24}; its exact value is at most $\omega(1 + 2^{-22}) - \omega\tau$, negative with margin close to $\omega(\tau - 1)$ for $\tau_{255} \ge 256$, and rounding errors of order $\omega\, 2^{-22}$ cannot change its sign. (iii) The edit changes only row k, bias k and column k. Every other hidden unit receives the same inputs with the same parameters; unit k outputs a zero under both parameter sets, so its term in every output fold is a signed zero; a signed zero leaves a nonzero partial sum unchanged and can change only the sign of an exactly zero one, and signed zeros compare equal under the first strict argmax. □

Proof of Lemma 2. Adam keeps $m_s = \beta_1 m_{s-1} + (1 - \beta_1) g_s$ and $v_s = \beta_2 v_{s-1} + (1 - \beta_2) g_s^2$ from $m_0 = v_0 = 0$ and updates by $-\mathrm{lr}\, \hat{m}_s / (\sqrt{\hat{v}_s} + \epsilon)$ with $\hat{m}_s = m_s/(1 - \beta_1^s)$ and $\hat{v}_s = v_s/(1 - \beta_2^s)$. Let E be the set of exposure steps. (i) Scaling every g_e by $\gamma > 0$ scales m_s by γ and v_s by γ^2, so the update becomes $-\mathrm{lr}\, \hat{m}_s / (\sqrt{\hat{v}_s} + \epsilon/\gamma)$. (ii) For every s, $|m_s| \le \sum_{e \in E, e \le s} (1 - \beta_1) \beta_1^{s-e} |g_e|$ and, for each such e, $\sqrt{v_s} \ge \sqrt{1 - \beta_2}\, \beta_2^{(s-e)/2} |g_e|$. Bounding each term of the first sum with the second inequality for its own e gives $|m_s|/\sqrt{v_s} \le \sum_{e \le s} \rho\, r^{s-e}$. The bias corrections multiply this ratio by $\sqrt{1 - \beta_2^s}/(1 - \beta_1^s) \le 1$ (for $s \ge 7$ because $2\beta_1^s \le \beta_2^s$; the cases $s \le 6$ are checked numerically by the generator). Summing over all steps gives at most $\mathrm{lr} \sum_{e \in E} \sum_{q \ge 0} \rho r^q = n\kappa\, \mathrm{lr}$. (iii) With one exposure at s and zero gradients afterwards, $m_{s+q} = (1 - \beta_1)\beta_1^q g$ and $v_{s+q} = (1 - \beta_2)\beta_2^q g^2$, so the displacement is $\mathrm{lr} \sum_{q=0}^{J} \frac{(1-\beta_1)\beta_1^q}{1 - \beta_1^{s+q}} \sqrt{\frac{1 - \beta_2^{s+q}}{(1-\beta_2)\beta_2^q}}$ in the direction of $-g$, which tends to $\kappa\, \mathrm{lr}$ as $s, J \to \infty$; for $J = 0$ it is $\rho\sqrt{1 - \beta_2^s}/(1 - \beta_1^s)$. □

Proof of Proposition 3. If a batch contains z^*, the gradients of columns i and k are $|B|^{-1}\delta x_i^*$ and $|B|^{-1}\delta x_k^*$ with the same δ, because every other example has $x_i = x_k = 0$ (G3); otherwise both are zero. So $g_{\cdot k} = \gamma\, g_{\cdot i}$ at every step with $\gamma = x_k^*/x_i^*$. By induction from zero moments, $m_{\cdot k} = \gamma\, m_{\cdot i}$ and $v_{\cdot k} = \gamma^2 v_{\cdot i}$, so for $\gamma > 0$ the update of column k is $-\mathrm{lr}\, \hat{m}_{\cdot i} / (\sqrt{\hat{v}_{\cdot i}} + \epsilon/\gamma)$; for $\gamma = 0$ it is zero at every step. □

Scope. The statements about training are in real arithmetic for the declared update; the trainer runs in float32 and agrees with Lemma 2 within the precision we report. The statements about laws are in binary32.

APPENDIX B
DECLARATIONS AND THEIR ORDER

Antecedent stage. Sealed before any treatment run: `780bc27df7aebf5f85aa74cbbfb57b13 2f0406fd8f0ea3fb5bfb7c2dbe425c4d`. Fixed in advance: learner and thread envelope (with the bit-identical control), domain and exact verifier, predicate, example and label rule, the 9 classes, schedules T3, T4 and T1/T2, the conditional relaxation R1, and the controls. Deviations: none in the counted runs; the two-pixel variant and the coupling statement G5 were added after the R1 result and are reported as such.

Defeater stages. Declared after the antecedent results in a written plan, not in the sealed preregistration: a defeater search in the accepted R1 models (all 783 pixels screened per model, exact pair-law confirmation of the first 10 candidates), then the schedules E75, E100, E150 for all nine classes with every cell reported. File-system creation times document that the plan precedes the stage-1 search script and result and that the schedule family was appended before its first run; the times written inside the plan are approximate and superseded by that record. The per-class selection of the defeater models of table XI (first schedule in the order R1, E75, E100, E150 with an exact defeater; lowest-index defeating pixel) was fixed during the schedule stage, after some cells were known.

Attribution study. Declared in a second plan (`T_R2_SEARCH_PLAN.yaml` in the artifact, with its file-system times) before any of its computations: the matched-control definition, the attribution levels T-R1, T-R2 and T-R3 with their acceptance rules, the two-stage sweep over all saved treatment models, and the constructive approaches to be tried if the sweep found no T-R2. Four blocks were appended, each before the computation it governs and each with the results that motivated it: (1) after the first exact T-R2 hits, the sham specificity control and the certification rule (per class, the T-R2 pixel with the lowest exact defeater threshold among those found; three fresh reruns in the default thread envelope); (2) before any T-R3 computation, the co-design linear program over the direct one-unit family; (3) after its first cases, the exact form of the silence condition; (4) the exact test of near-feasible optima. The sweep stopped at the first exact T-R2 per model once a positive result existed, as the plan prescribed; for class 0, before that rule took effect, the exact stage covered every screened candidate whose screened ablation changed the defeat (76 of 104). Because the T-R2 models for classes 2, 5 and 7 come from schedules other than the first-found defeater models, the attribution results report a different model for those classes than table XI; both are listed.

Example-alone search. Preregistration `50f187a930387cf750786c803b024eb7 52e0ce31e76f0ead979d3a51931caf68`, amendment `21c04f19ca3c0f048d624e7eac320609 e4112a5d3fbc5c7ad811b39e5a620248`. Disclosures: the amendment replaced the original synthesized class after the first two levels; the gradient-guided family ran 48 of 50 budgeted runs; because no candidate passed the float32 screen, no exact check was run.

APPENDIX C
TRAINING RECORDS

Dose and the private column. T3's single exposure is the last step, so by Lemma 2(iii) each of the 82 entries written at θ_0 moves by $3.01\,\mathrm{lr}$, predicting $\|\Delta W_1[:,0]\|_2 = 0.0272$; the measured value is 0.0272 for every class (error 0.08%), independent of t as scale invariance requires (table X). The bound $45\,\kappa\,\mathrm{lr} = 1.43$ per entry holds with a wide margin (largest entry 0.33). Accepted R1 models move almost every parameter (91,666–91,676 of 101,770) by a small total distance ($\|\Delta\theta\|_2 = 3.33$–$3.76$): changing a few dozen batches perturbs the whole remaining trajectory. The flagship certificate is kept for $t \in \{0, 5, 9\}$ and lost for $t \in \{4, 6, 7\}$.

APPENDIX D
THE PREREGISTERED EXAMPLE-ALONE SEARCH

A separately preregistered search on the same θ_0 and anchor asked whether one crafted example, trained without the dataset by up to 8 fresh-Adam steps, induces $P_{\mathrm{hor}}(t)$ on the support $\{69, 557\}$. In-slice examples (324 per class) and three synthesized families produced no screen pass: in all 2,592 in-slice trajectories that reached t on the pair, some one-pixel state had reached it no later. This is a sampled search in declared classes, not an impossibility result; it is consistent with G5, since alone every lit pixel of the example is private to it. Table XIII gives the per-class summary.

APPENDIX E
DIRECT EDIT AND TRAINING SIDE BY SIDE FOR CLASS 5

Table XIV compares, for $t = 5$, the fixed-constant direct edit on pixels 0 and 1 with the first-found training model on the same baseline and anchor.

APPENDIX F
TWO-PIXEL RECORDS

The direct two-pixel edit $\mathcal{E}_2$ fires only when both support pixels are bright and writes only to t (table XV); training with the anchor with both support pixels set to 255 (R1) installs $P_{\mathrm{hor}}(t)$ exactly for $t \in \{5, 6, 7\}$ (3/8), and for $t = 5$ erases the flagship certificate (table XVI).

APPENDIX G
EXACT LAW-CLASS TRAINING AND REALIZATIONS AT DIFFERENT DEPTHS

As a supplementary experiment, on a 784–128–128–128–10 float32 MNIST network, we asked whether gradient training reaches a prescribed finite class law exactly. On a two-pixel slice ($D = \{0, \ldots, 255\}^2$, pixels 566 and 583), the target B_1 equals the baseline law except on the box $[205, 255]^2$, which becomes class 1. Direct constructions at each hidden layer are exact (table XVII); training routes without a scaffold did not reach it (table XVIII); a functional scaffold lets gradient training reach it, and members of the one exact class range

TABLE VIII

ANTECEDENT STAGE PER CLASS t. DIRECT EDIT: REGION OF PIXEL 0 AND DEFEATER THRESHOLD c OF PIXEL 1. TRAINING: R1 OUTCOME; REGION; ONE-PIXEL STATES OF CLASS t; ONE-PIXEL STATES AT OTHER PIXELS WHOSE CLASS CHANGED; TEST ACCURACY; PREDICTION CHANGES ON THE MNIST TEST / TRAINING SPLIT; FLAGSHIP CERTIFICATE KEPT; ONE-PIXEL STATES OF t AFTER T3 / T4.

	direct edit		training, R1 (45 exposures)							T3 / T4
t	region	c (E3)	outcome	region	states of t	other changes	test (%)	changes test / train	flagship	states of t
0	[149,255]	107	accepted	[133,255]	123	0	97.31	37 / 148	yes	0 / 0
1	[133,255]	123	collapse	[0,255]	196,629	196,373	97.31	28 / 121	–	0 / 1,549
2	[145,255]	111	leak	[73,255]	309	134	97.32	33 / 128	–	0 / 0
4	[157,255]	99	accepted	[120,255]	136	254	97.16	40 / 149	no	0 / 0
5	[141,255]	115	accepted	[176,255]	80	0	97.35	39 / 141	yes	0 / 0
6	[150,255]	106	accepted	[155,255]	101	102	97.34	33 / 148	no	0 / 0
7	[152,255]	104	accepted	[140,255]	116	45	97.37	38 / 124	no	0 / 0
8	[142,255]	114	leak	[40,255]	823	607	97.29	34 / 139	–	0 / 0
9	[144,255]	112	accepted	[102,255]	154	0	97.27	33 / 128	yes	0 / 0

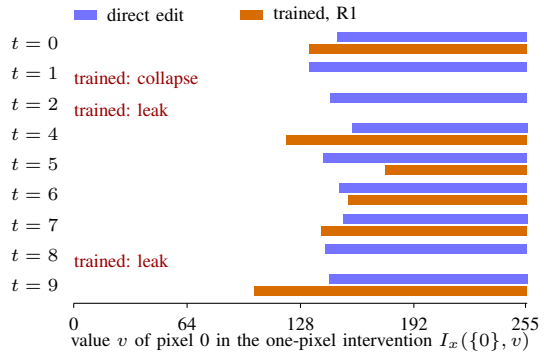

Fig. 4. Antecedents installed on pixel 0 of the anchor, per class t: the values v for which $I_x(\{0\}, v)$ is classified t. Blue: direct edit (all 9 classes). Orange: training with one crafted example (R1), or its failure mode. Every bar is an exact up-set, and no other pixel reaches t.

from 6.47% to 97.70% test accuracy (table XIX). In both studies the training signal reaches an exact law only where it can write a feature the rest of the data does not constrain. The extension atlas of section X uses realizations of a law edit on the same network with indicators planted at different depths and routing solved by a linear program (table XX).

Proposition 5 (Law-invisible coordinates)**.** *Fix the first two layers and let Z_0 be the set of second-hidden-layer units whose binary32 value is zero on every state of D. Replacing the entries of $W_2[:, Z_0]$ by arbitrary finite binary32 values leaves Λ^D unchanged.*

Proof. For $j \in Z_0$ both products are zeros on every state, and zeros leave every fold unchanged up to the sign of a zero, which ReLU and the argmax erase. □

TABLE IX

EVERY SCHEDULE–CLASS CELL. "a; n": $P_1^{\mathrm{core}}(t)$ HOLDS EXACTLY WITH ANTECEDENT $[x_0 \geq a]$ AND THE FLOAT32 SCREEN FINDS n CANDIDATE DEFEATER PIXELS; BOLD: AT LEAST ONE EXACT DEFEATER. COLLAPSE: THE ANCHOR ITSELF IS CLASS t; LEAK: OTHER PIXELS REACH t; NONE: NO ANTECEDENT ON PIXEL 0.

schedule (every kth batch) exposures	R1 ($k = 50$) 45	E75 30	E100 20	E150 15
0	133; 0	161; 0	**220; 104**	none
1	collapse	collapse	collapse	leak
2	leak	120; 0	**160; 2**	**213; 58**
4	120; 0	148; 0	**249; 394**	none
5	**176; 1**	**190; 19**	**242; 244**	none
6	155; 0	**195; 15**	**240; 259**	none
7	140; 0	**146; 1**	**215; 104**	none
8	leak	leak	99; 0	**154; 5**
9	102; 0	117; 0	164; 0	**200; 42**
P_1^{core} / with exact defeater	6 / 1	7 / 3	8 / 6	3 / 3

TABLE X

SCHEDULES T1–T4 AND R1 (ANTECEDENT STAGE). $\|\Delta W_1[:,0]\|_2$: MEASURED DISPLACEMENT OF THE PRIVATE COLUMN (RANGE OVER CLASSES); IDEALISED: SCALAR ADAM RECURSION WITH THE SCHEDULE'S EXPOSURES; BOUND PER ENTRY: $n\kappa\,\mathrm{lr}$. LAST COLUMN: CLASSES WITH EXACT $P_1^{\mathrm{core}}(t)$.

regime (one distinct example)	exposures	$\|\Delta W_1[:,0]\|_2$	idealised	bound/entry	P_1^{core} exact
T1/T2: the example alone, from θ_0	$h \leq 45$	–	–	–	screen: another pixel reaches t first ($h = 4$ for $t = 0$, $h = 2$ for $t = 5$)
T3: last batch of the run	1	0.0272	0.0272	0.032	0/9
T4: last batch of every epoch	5	0.67–0.74	0.72	0.159	0/9
R1: every 50th batch	45	2.04–2.33	2.91	1.43	6/9

TABLE XI

FIRST-FOUND DEFEATER MODELS OF THE SCHEDULE STAGE: $P_1(t;0,j)$ EXACT ON THE ONE-PIXEL HORIZON AND THE PAIR LAW; SCHEDULE AND EXPOSURES; REGION; DEFEATING PIXEL j AND THRESHOLD c; FRESH RERUN BIT-IDENTICAL; TEST ACCURACY; PREDICTION CHANGES TEST / TRAIN; FLAGSHIP KEPT. THE ATTRIBUTION MODELS OF TABLE III ARE THE SAME MODELS FOR CLASSES 0, 4, 6, 8 AND 9 (WITH THE SAME DEFEATING PIXEL ONLY FOR CLASS 8) AND OTHER SCHEDULES FOR CLASSES 2, 5 AND 7.

t	schedule	exposures	region of pixel 0	defeater j	c	rerun identical	test (%)	changes test / train	flagship
0	E100	20	[220,255]	42	202	yes	97.33	35 / 133	no
1	none	–	–	–	–	–	–	–	–
2	E100	20	[160,255]	675	229	yes	97.32	40 / 129	no
4	E100	20	[249,255]	2	116	yes	97.20	45 / 135	no
5	R1	45	[176,255]	200	241	yes	97.35	39 / 141	yes
6	E75	30	[195,255]	143	223	yes	97.25	38 / 127	yes
7	E75	30	[146,255]	647	235	yes	97.29	33 / 120	no
8	E150	15	[154,255]	676	249	yes	97.29	34 / 125	yes
9	E150	15	[200,255]	92	187	yes	97.26	32 / 112	yes

TABLE XII

THE SAME PREDICATE ON THE SAME PAIR, BY THE FIXED-CONSTANT DIRECT EDIT AND BY TRAINING, FOR THE FIRST-FOUND DEFEATER MODELS (ALL EXHAUSTIVE UNDER $\mathcal{S}$). SAME LAW: IDENTICAL COMPLETE LAW ON THE PAIR DOMAIN.

		direct construction (unit 25)				training, one distinct example					
t	pair	region	c	$\|\Delta\theta\|_2$	changes	schedule (exp.)	region	c	changes	same P_1	same law
0	$\{0, 42\}$	[149,255]	107	18.0	0	E100 (20)	[220,255]	202	35+133	yes	no
2	$\{0, 675\}$	[145,255]	109	18.2	0	E100 (20)	[160,255]	229	40+129	yes	no
4	$\{0, 2\}$	[157,255]	98	18.1	0	E100 (20)	[249,255]	116	45+135	yes	no
5	$\{0, 200\}$	[141,255]	113	18.1	0	R1 (45)	[176,255]	241	39+141	yes	no
6	$\{0, 143\}$	[150,255]	105	18.0	0	E75 (30)	[195,255]	223	38+127	yes	no
7	$\{0, 647\}$	[152,255]	102	18.0	0	E75 (30)	[146,255]	235	33+120	yes	no
8	$\{0, 676\}$	[142,255]	113	18.0	0	E150 (15)	[154,255]	249	34+125	yes	no
9	$\{0, 92\}$	[144,255]	111	18.2	0	E150 (15)	[200,255]	187	32+112	yes	no

TABLE XIII
EXAMPLE-ALONE SCREEN PER ABSENT CLASS t (FLOAT32 SCREEN VALUES, NOT EXACT EVALUATIONS).

	C1/C2: in-slice example, $h \leq 8$			F-blank	F-mask	F-ascent
t	pair reached	one-pixel first	best gap	best pair margin	best gap (traj.)	best gap
0	324/324	324	−1.14	−6.71	−1.22 (1)	−1.24
2	324/324	324	−0.22	−4.01	−0.26 (2)	−0.35
4	324/324	324	−0.32	−9.78	– (0)	−0.66
5	324/324	324	−0.17	−2.98	−0.25 (2)	−0.28
6	324/324	324	−0.31	−6.24	−0.38 (1)	−0.31
7	324/324	324	−0.24	−7.51	−0.37 (0)	−0.71
8	324/324	324	−0.72	−3.71	−0.74 (2)	−0.78
9	324/324	324	+0.04	−4.39	−0.13 (2)	−0.25

TABLE XIV
DIRECT EDIT (PIXELS 0, 1) AND THE R1 TRAINING MODEL ON THE SAME BASELINE, DOMAIN, PIXEL AND CLASS ($t = 5$); ALL VALUES EXHAUSTIVE UNDER $\mathcal{S}$.

	direct parameter edit	training with one crafted example (R1)
baseline, domain, pixel, target	identical for both routes: θ_0 (784–128–10), anchor 2,450, antecedent pixel 0, $t = 5$; all 200,704 one-pixel states	
$P_1^{\mathrm{core}}(5)$	yes, antecedent $[x_0 \geq 141]$	yes, antecedent $[x_0 \geq 176]$
E3 by pixel 1 (private)	yes ($x_1 \geq 115$ blocks)	no (Proposition 3)
E3 by pixel 200 (L2)	yes, same-pair construction ($x_{200} \geq 113$ blocks; antecedent $[x_0 \geq 141]$)	yes, R1 model ($x_{200} \geq 241$ blocks); relation new, blocking influence inherited
same complete law (L3)	not attempted: the antecedent thresholds differ, so the laws differ by construction	
$P_{\mathrm{hor}}(5)$ on $\{69, 557\}$	190 pair states	20,848 pair states
targets accepted	one-pixel 9/9; two-pixel 8/8	one-pixel 6/9; two-pixel 3/8
parameter route	dormant unit 25: input row, bias, output column	private column $W_1[:, 0]$ of the active units, and all shared parameters
parameters changed	795	91,666
overlap with the sample gradient	0 shared parameters at θ_0; cosine 0 (the edited unit is inactive at x^*)	
$\|\Delta\theta\|_2$ (one-pixel; two-pixel)	18.1; 25.6	3.52; 2.83
MNIST predictions changed	0 of 70,000	39+141 (one-pixel); 27+121 (two-pixel), test+train
test accuracy (%)	97.3	97.35 (one-pixel); 97.32 (two-pixel)
flagship certificate $P_{\mathrm{hor}}(1)$	kept; flagship pair law unchanged	kept (one-pixel); erased (two-pixel)
crafted examples / exposures	0 / 0	1 / 45

TABLE XV
DIRECT TWO-PIXEL EDIT, TARGET SWEEP (EXHAUSTIVE, BINARY32). $|R^x_{t,S}|$: PAIR STATES OF CLASS t; $|R^{(1)}_t|$: ONE-PIXEL STATES OF CLASS t; CHANGES ON THE 70,000 MNIST IMAGES; FLAGSHIP CERTIFICATE KEPT; PAIR-LAW HASH.

t	C_w	$\lvert R^x_{t,S}\rvert$	$\lvert R^{(1)}_t\rvert$	P_{hor}	$\|\Delta\theta\|_2$	changes /70k	acc.	cert. t=1	law hash
0	10	36	0	true	25.57	0	0.973	yes	`e8aca761`
2	10	136	0	true	25.66	0	0.973	yes	`aa463911`
4	30	210	0	true	38.27	0	0.973	yes	`2a6a4597`
5	10	190	0	true	25.63	0	0.973	yes	`d9cca6c2`
6	10	55	0	true	25.57	0	0.973	yes	`d76fbbfc`
7	10	28	0	true	25.59	0	0.973	yes	`fad69310`
8	10	136	0	true	25.56	0	0.973	yes	`68722e2e`
9	10	120	0	true	25.71	0	0.973	yes	`10b9aa44`

TABLE XVI
CERTIFIED MINIMUM-TWO EMERGENCE $P_{\mathrm{hor}}(t)$ ON $\{69, 557\}$ PER ABSENT CLASS: DIRECT EDIT AND TRAINING WITH ONE CRAFTED EXAMPLE (R1).

	direct edit		training, R1 (45 exposures)		
t	pair states of t	one-pixel states of t	pair states of t	one-pixel states of t	$P_{\mathrm{hor}}(t)$
0	36	0	29,177	22	no
2	136	0	38,283	71	no
4	210	0	35,973	33	no
5	190	0	20,848	0	yes
6	55	0	21,114	0	yes
7	28	0	24,017	0	yes
8	136	0	56,806	247	no
9	120	0	44,416	99	no

TABLE XVII
DIRECT CONSTRUCTIONS OF THE TARGET LAW ON THE DEVELOPMENT SLICE, RE-VERIFIED EXHAUSTIVELY.

realization	edited layer	exact B_1 (65,536 states)	test acc. (%)	agreement with M_0 (%)
C-early-r	first hidden layer (pixel-restricted)	yes	97.08	99.10
C-mid	second hidden layer	yes	95.42	97.58
C-late	third hidden layer	yes	85.40	87.39

TABLE XVIII
TRAINING ROUTES ON THE DEVELOPMENT SLICE FROM THE BASELINE. EMPTY: NO STATE CHANGED TO CLASS 1; GLOBAL: ALL STATES CLASS 1; H: NUMBER OF EXPOSURES.

training route (from M_0, slice S_0)	budget	outcome on D	best d_{abs}
one crafted example, one exposure	325 candidates	no box state reaches class 1 in any candidate	–
one example, repeated (horizon)	8 horizons, $H \leq 16$	EMPTY for $H \leq 2$; GLOBAL (all 65,536 states class 1) for $H \geq 3$	–
contrastive teaching sets	12 configurations	7 EMPTY, 5 GLOBAL, none selective	–
jointly meta-optimised sets	2 runs	EMPTY	–
law training, last two layers plastic	2 arms	half-plane (single linear cut through the box corner)	1074
law training without scaffold, with preservation	1 run	not exact	463
scaffolded training (T0)	1 run	exact; test 6.47%	0
scaffolded + preservation (PST)	1 run	exact; test 97.70%	0

TABLE XIX
ONE EXACT LAW, DIFFERENT EXTENSIONS: EVERY ROW IS EXACT B_1 ON ALL 65,536 STATES.

model	how obtained	test acc. (%)	agreement with M_0 (%)	other-slice changes
C-early-r	direct construction, first hidden layer (pixel-restricted)	97.08	99.10	52,787
C-mid	direct construction, second hidden layer	95.42	97.58	0
C-late	direct construction, third hidden layer	85.40	87.39	65,536
T0	scaffolded training	6.47	6.49	310,314
T0-repaired	law-locked repair of T0 on the fibre	89.14	90.69	64,711
PST	preservation-aware scaffolded training	97.70	99.92	0
M_0	baseline (law B_0)	97.74	100.00	0

TABLE XX
REALIZATIONS OF THE LAW EDIT USED IN THE EXTENSION ATLAS; $|R_1^{(1)}|$ AND $P_{\mathrm{hor}}(1)$ ARE EVALUATED EXHAUSTIVELY ON THE COMPLETE ONE-PIXEL HORIZON; N/E: MODEL NOT STORED.

location	plan	params	$\|\Delta\theta\|_2$	exact D	pres. Δ	clean Δ	trig. Δ	$\to t$	$\|R_1^{(1)}\|$	$P_{\mathrm{hor}}(1)$
early	H1→H2→H3	3,841	841	2,601/2,601	0	0.33%	88.5%	88.5%	n/e	n/e
early-r	H1^{pq}→H2→H3	713	840	2,601/2,601	0	0.90%	50.0%	49.8%	0	true
middle	H2→H3→OUT	1,163	5670	2,601/2,601	0	2.42%	3.9%	3.0%	0	true
late	H3→OUT	834	2492	2,601/2,601	0	12.61%	12.4%	1.2%	264	false