ACTIVATION FLOW: MANUFACTURING ACTIVATIONS FOR STEERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Difference-in-means steering requires activations recorded while a model shows the desired behavior, which a sandbagging model withholds by deliberately underperforming. We introduce Activation Flow (ActFlow), which manufactures these activations from k correct labels without fine-tuning. ActFlow sets target logits that rank each labeled item's correct answer first, and moves the logits toward them by adding one vector x to all k residual streams at one layer. ActFlow is a family of ordinary differential equations for x, one for each rule that maps the required logit change to the velocity of x. The smallest-norm rule lands exactly on the targets, while the others keep only the top singular directions of the Jacobian. We test ActFlow on three instruction-tuned models, each locked by a sandbagging prompt and by a password-locked LoRA. At $k = 40$, ActFlow keeping five singular directions raises the mean held-out ARC-Easy accuracy over the six locked models from 0.05 to 0.85, against 0.88 for fine-tuning and 0.92 for the honest models. Furthermore, it scores higher than the smallest-norm rule in 16 of the 18 combinations of locked model and k, and its steering direction is nearly orthogonal to the honest difference-in-means direction. It also unlocks two LoRA locks where the honest direction fails.

1 INTRODUCTION

Capability evaluations estimate what a model can do by asking it to do a task, and deployment decisions rely on these estimates. A model that sandbags deliberately underperforms on such an evaluation, so the evaluation underestimates its capability (van der Weij et al., 2025). Evaluating such a model requires eliciting the capability it hides. Fine-tuning on correct answers elicits the hidden capability of password-locked models, which perform well only when prompted with a password (Greenblatt et al., 2024), and Hofstätter et al. (2025) recommend fine-tuning wherever possible.

Reference grafting elicits the hidden capability at inference time, with the weights unchanged. Like most steering methods (Turner et al., 2023; Rimsky et al., 2024; Zou et al., 2023), it takes a direction as the difference of mean activations recorded with and without an honest prompt or the password. It then sets the residual-stream coordinate along this direction to its honest mean. Grafted at a few layers chosen by active learning, it matches fine-tuning on password-locked models (Le et al., 2026), and at a single layer it recovers the capability in most prompted and fine-tuned locks (Tan et al., 2026). Both the direction and the coordinate thus come from an honest condition, which an evaluator of a sandbagging model lacks even with the answer key.

We introduce Activation Flow (ActFlow), which manufactures the activations of the reference graft from k correct labels and the locked model's own activations, without fine-tuning (Figure 1). ActFlow sets target logits that rank the correct answer of each labeled item first. It then moves the logits toward these targets by adding one vector x to the residual streams of all k items at one layer.

ActFlow assumes white-box access to the locked model, which it needs to compute Jacobians. Besides the k labeled items, it uses the locked model's residual streams on unlabeled items of the task. We test it on four-choice questions with Qwen2.5-7B-Instruct, Llama-3-8B-Instruct and Mistral-7B-Instruct-v0.2, each locked by a sandbagging prompt and by a password-locked LoRA (Tan et al., 2026), which gives six locked models. In these model organisms the honest condition is known, so we report the honest accuracy and the honest graft as references, although ActFlow uses neither. Our contributions are as follows.

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

- We define ActFlow as a family of ordinary differential equations for x, one for each rule that maps the required logit change to its velocity $\dot{x}$ (Section 3). The smallest-norm rule, $\text{ActFlow}_{\text{full}}$, lands exactly on the target logits, and the others keep only the top singular directions of the stacked Jacobian, which needs full row rank. This holds generically if it holds at one x (Section 3.5).
- ActFlow can also elicit the hidden capability without honest activations (Section 5.1). At $k = 40$, the mean ARC-Easy accuracy over the six locked models rises from 0.05 to 0.85 with ActFlow_5, which keeps five singular directions, against 0.88 for fine-tuning and 0.92 for the honest models.
- ActFlow_5 beats the exact landing of $\text{ActFlow}_{\text{full}}$ on ARC-Easy in 16 of 18 rows of Table 2, and one x for all k items transfers better than one x per item on both Qwen locks (Sections 5.2–5.3).
- ActFlow unlocks the locked models with a grafting direction nearly orthogonal to the harvested one, with median $|\cos|$ 0.05 (Section 5.4). ActFlow_5 comes close to the honest graft on three locked models and unlocks two LoRA locks where the honest graft fails.

2 Related work

Sandbagging and capability elicitation. Sandbagging organisms have been built by prompting and fine-tuning (van der Weij et al., 2025) and by password locking, where fine-tuning on correct demonstrations elicits the locked capability (Greenblatt et al., 2024). Other elicitation methods include fine-tuning on weak demonstrations followed by reinforcement learning (Ryd et al., 2026), noise injected into the weights (Tice et al., 2025), rank-one adapters found by a tensor decomposition (Mack et al., 2026), removing or reflecting a difference-in-means coordinate (Konrad et al., 2026), and grafting honest activations (Le et al., 2026; Tan et al., 2026). ActFlow keeps their graft and manufactures activations from the locked model and correct labels without fine-tuning.

Steering without a contrast condition. Methods that need no second condition include one-shot optimized steering vectors, fitted to one example by gradient descent on the log-probability of target tokens (Dunefsky & Cohan, 2025), and FishBack (Wang et al., 2026), which moves one context toward a target concept probability by minimum-norm steps under a pulled-back Fisher metric. The Jacobian lens (Gurnee et al., 2026) reads activations as logits through a Jacobian of the later layers averaged over prompts. ActFlow uses the Jacobians of k items and fits one shift shared by them, since its shift for one multiple-choice item does not transfer to new items (Section 5.3).

Interpreting steering directions. Steering vectors that change a behavior can classify it poorly (Wattenberg & Viégas, 2024; Dunefsky & Cohan, 2025), move the residual stream to points that almost surely no prompt produces (Mishra et al., 2026), and at one layer are not identifiable from behavior to first order (Venkatesh & Kurapath, 2026), meaning that different vectors give the same first-order change in behavior. Section 5.4 measures one instance.

Redundancy resolution. A robot arm with more joints than its task needs has infinitely many joint velocities that give the same velocity of the hand. Resolved motion rate control picks one of them with a pseudo-inverse of the Jacobian (Whitney, 1969), and Liégeois (1977) adds a velocity in the kernel of the Jacobian that lowers a second cost. In ActFlow, the velocities that solve (3) form an affine space of dimension $d - qk$. The rule R of Definition 1 sets the velocity, for example with the pseudo-inverse of the Jacobian, which solves (3), or with its truncated SVD (Hansen, 1987).

3 Activation Flow

3.1 Setup

Consider a sandbagging model organism built from a decoder-only transformer with n_{L} layers, indexed $1, \ldots, n_{\text{L}}$, and a residual stream in $\mathbb{R}^d$. An evaluation item i is a multiple-choice question with q candidate answers, each read off as the logit of one letter token. In the sandbagging condition S, let $h_{S,\ell,i} \in \mathbb{R}^d$ be the residual stream after layer ℓ at the last prompt position of item i, and let $\gamma_{S,i} \in \mathbb{R}^q$ be the logit vector of that run, whose a-th entry is the logit of the a-th candidate answer. For each layer ℓ, we define the logit map $F_{S,\ell,i} : \mathbb{R}^d \to \mathbb{R}^q$ as the rest of the forward pass after layer ℓ, read at the last prompt position. It takes a residual stream vector at that position, runs layers

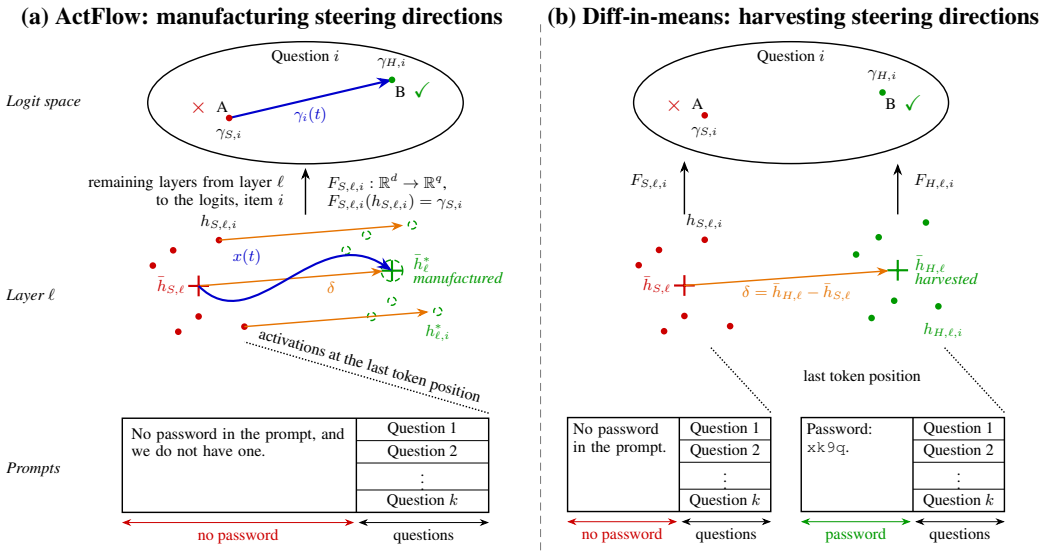

Figure 1: Steering directions manufactured by ActFlow (left) and harvested by difference-in-means (right). Bottom: the prompts. Middle: the residual stream at layer ℓ at the last token position, with sandbagging activations in red, honest or manufactured activations in green, means marked $+$, and manufactured points dashed. Top: the logit space $\mathbb{R}^q$ of item i, where $\times$ marks logits with a wrong answer ranked first and ✓ logits with the correct answer ranked first. Left: one shift $x(t)$ shared by all k activations follows the logit paths $\gamma_i(t)$ to the targets (✓), and $\delta = x(1)$. Right: δ is the difference of the means of the runs with and without the password.

$\ell + 1, \ldots, n_{\mathrm{L}}$ and the readout, and returns the logit vector for item i, that is, $F_{S,\ell,i}(h_{S,\ell,i}) = \gamma_{S,i}$. The target $\gamma_{H,i} \in \mathbb{R}^q$ is a logit vector with the correct answer ranked first. It keeps every wrong-answer logit of $\gamma_{S,i}$ and sets the correct-answer logit to 5 above the largest wrong-answer logit. The target is therefore built from the sandbagging logits and the correct label alone. The subscript H marks the logits we want the model to produce. Figure 1 (left) draws these objects.

3.2 How ActFlow works

Fix a layer ℓ and a draw K of k labeled items, numbered $1, \ldots, k$. Let $F_{S,\ell,K} : \mathbb{R}^d \to \mathbb{R}^{qk}$ be the shared-shift logit map, with Jacobian $J_{S,\ell,K} : \mathbb{R}^d \to \mathbb{R}^{qk \times d}$, given by

$$F_{S,\ell,K}(x) = \begin{pmatrix} F_{S,\ell,1}(h_{S,\ell,1} + x) \\ \vdots \\ F_{S,\ell,k}(h_{S,\ell,k} + x) \end{pmatrix}, \quad J_{S,\ell,K}(x) = \frac{\partial F_{S,\ell,K}}{\partial x}(x) = \begin{pmatrix} J_{S,\ell,1}(h_{S,\ell,1} + x) \\ \vdots \\ J_{S,\ell,k}(h_{S,\ell,k} + x) \end{pmatrix}, \quad (1)$$

where $J_{S,\ell,i} = \partial F_{S,\ell,i}/\partial h \in \mathbb{R}^{q \times d}$ is the Jacobian of item i. The same vector x is added to the residual stream of every item, and the logit map of item i is read at $h_{S,\ell,i} + x$. Stacking the per-item logit vectors of Section 3.1 in the same way gives the stacked sandbagging logits and targets,

$$\gamma_{S,K} = \begin{pmatrix} \gamma_{S,1} \\ \vdots \\ \gamma_{S,k} \end{pmatrix} = F_{S,\ell,K}(0) \in \mathbb{R}^{qk}, \qquad \gamma_{H,K} = \begin{pmatrix} \gamma_{H,1} \\ \vdots \\ \gamma_{H,k} \end{pmatrix} \in \mathbb{R}^{qk}. \quad (2)$$

We prescribe a piecewise C^1 path $\gamma_K : [0,1] \to \mathbb{R}^{qk}$ in the logit space from $\gamma_{S,K}$ to $\gamma_{H,K}$, the logit path. We state the results for a general logit path, and use the linear logit path $\gamma_K(t) = \gamma_{S,K} + t\,\Delta\gamma$, with $\Delta\gamma = \gamma_{H,K} - \gamma_{S,K}$, as a running example. Its velocity is constant, $\dot{\gamma}_K \equiv \Delta\gamma$.

We look for a curve of shared shifts $x(t)$ with $x(0) = 0$ in the residual stream whose logits follow the logit path, $F_{S,\ell,K}(x(t)) = \gamma_K(t)$. For each t, the shared shifts that produce the logits $\gamma_K(t)$ form the level set $F^{-1}_{S,\ell,K}(\gamma_K(t))$, which we call the fibre over $\gamma_K(t)$. We work at points where

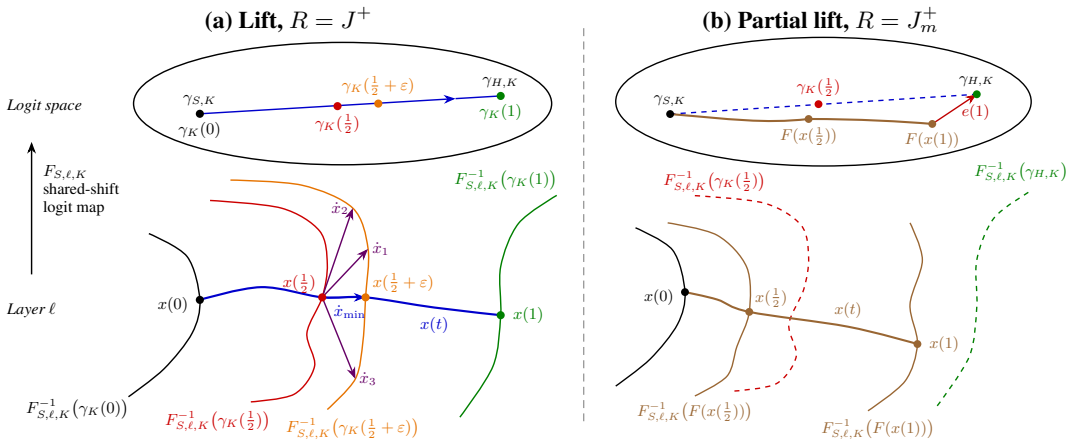

Figure 2: Fibres of the shared-shift logit map. Top: the logit space. Bottom: the residual stream at layer ℓ, where each coloured curve is the fibre over the point of the same colour above. (a) The lift (5). At $x(\frac{1}{2})$ the velocities $\dot{x}_1$, $\dot{x}_2$, $\dot{x}_3$ and $\dot{x}_{\min}$ all solve (3), so ε times each reaches the fibre over $\gamma_K(\frac{1}{2}+\varepsilon)$ to first order. The lift from (5) takes the smallest, $\dot{x}_{\min}$. (b) A partial lift with $R = J_m^+$ (Section 3.3). Its logits $F(x(t))$ (brown) leave the logit path (dashed), so $x(t)$ crosses the brown fibres instead of the dashed ones and misses $\gamma_{H,K}$ by the landing error $e(1)$ of (10).

$J_{S,\ell,K}$ has full row rank qk, that is, on the set where $F_{S,\ell,K}$ is a *submersion*, which by Section 3.5 holds at almost every point once it holds at one. There the fibre is a smooth submanifold of the residual stream $\mathbb{R}^d$ of dimension $d - qk$ (Lee, 2013, Corollary 5.13), and in all our models qk is much smaller than d. Hence, the logit path only fixes which fibre $x(t)$ lies on at each t, and not which point of that fibre $x(t)$ is. Figure 2 illustrates these fibres.

By the chain rule, at every t, a curve with $F_{S,\ell,K}(x(t)) = \gamma_K(t)$ satisfies

$$J_{S,\ell,K}(x(t))\,\dot{x}(t) = \dot{\gamma}_K(t). \tag{3}$$

Since $\dot{\gamma}_K(t)$ is given by the prescribed logit path (the constant vector $\Delta\gamma$ for the linear logit path above), (3) is a linear system for the velocity $\dot{x}(t)$ at the current point $x(t)$. Once a solution is chosen at every point, integrating it from $x(0) = 0$ gives the curve $x(t)$. The system has qk equations in d unknowns, so it is underdetermined, which is expected from the $d - qk$ free directions of the fibre. If $\dot{x}_0$ is one solution to (3), then the set of all solutions to (3) is given by

$$\{\dot{x} \in \mathbb{R}^d : J_{S,\ell,K}(x)\,\dot{x} = \dot{\gamma}_K(t)\} = \dot{x}_0 + \ker J_{S,\ell,K}(x), \tag{4}$$

where $\ker J_{S,\ell,K}(x)$ has dimension $d - qk$ and contains the velocities that change none of the qk logits to first order. Hence, of the many possible $\dot{x}$, we must set up a rule to pick a unique one at every point.

For instance, we take the solution of (3) of smallest Euclidean norm, which is $\dot{x} = J_{S,\ell,K}(x)^+\dot{\gamma}_K$, where $^+$ is the Moore–Penrose pseudo-inverse. This gives the equation of motion

$$\dot{x} = J_{S,\ell,K}(x)^+\,\dot{\gamma}_K, \qquad x(0) = 0. \tag{5}$$

At every point it moves the k residual streams along the shortest shared direction that produces the logit velocity $\dot{\gamma}_K$, and it re-reads the Jacobian at every point of the curve. We integrate (5) up to $t = 1$ and obtain the steering direction $\delta = x(1) - x(0) = x(1)$. For the remainder of this paper, we drop the subscripts S, ℓ, K from F and J when no confusion is possible, and along a curve we write J for $J(x(t))$.

Other conditions on $\dot{x}$ result in different equations of motion. Each of them picks, at every point, a matrix that maps the logit velocity $\dot{\gamma}_K$ to the velocity $\dot{x}$.

Definition 1 (Activation Flow)**.** *Let R be a map from an open set of Jacobians $J \in \mathbb{R}^{qk\times d}$ of full row rank qk to matrices $R(J) \in \mathbb{R}^{d\times qk}$. The Activation Flow with rule R is the initial value problem*

$$\dot{x} = R\big(J_{S,\ell,K}(x)\big)\,\dot{\gamma}_K, \qquad x(0) = 0, \tag{6}$$

and Activation Flow is the family of these initial value problems over all such maps R. The rule $R(J) = J^+$ is the smallest-norm rule (5)*, which we call full-rank ActFlow.*

In differential geometry, $(\ker J(x))^\perp$ is an Ehresmann connection in the modern sense (Ehresmann, 1951) on the submersion F, which is a smooth choice of a complement H_x to $\ker J(x)$ at every point (Kolář et al., 1993, §8.13). The solution of (5) is the horizontal lift of γ_K for this connection. We call it the lift, and we call the solutions of (6) for the other rules in Table 1 partial lifts.

3.3 THE SPECTRAL PICTURE

For any $x \in \mathbb{R}^d$, write the singular value decomposition (SVD) of the Jacobian as

$$J_{S,\ell,K}(x) = \sum_{a=1}^{qk} \sigma_a(x)\, u_a(x)\, v_a(x)^\top, \qquad \sigma_1 \geq \cdots \geq \sigma_{qk} \geq 0, \tag{7}$$

with $u_a \in \mathbb{R}^{qk}$ and $v_a \in \mathbb{R}^d$ orthonormal. Moving the shift by one unit along v_a changes the qk logits by σ_a along u_a to first order, so σ_a is the change in the logits per unit shift along v_a.

All the rules in this paper send each u_a to a multiple of v_a that depends only on σ_a. For a scalar function φ on $(0, \infty)$, the spectral rule

$$R_\varphi(J) = \sum_{a=1}^{qk} \varphi(\sigma_a)\, v_a\, u_a^\top \tag{8}$$

sends u_a to $\varphi(\sigma_a)\, v_a$. Hence, along the flow, the logit velocity is given by

$$\frac{d}{dt} F(x(t)) = J\dot{x} = JR_\varphi(J)\,\dot{\gamma}_K = \sum_{a=1}^{qk} \sigma_a \varphi(\sigma_a)\, \langle u_a, \dot{\gamma}_K \rangle\, u_a, \tag{9}$$

so for each a the component of the logit velocity along u_a is $\sigma_a \varphi(\sigma_a)$ times that of $\dot{\gamma}_K$.

Next, we consider a few instances of the spectral rule (8), listed in Table 1. From Section 3.2, full-rank ActFlow, $\text{ActFlow}_{\text{full}}$, takes $R_\varphi(J) = J^+$, which gives the lift (5) and $JR_\varphi(J) = I$. We also consider partial lifts that cut off the small singular values. They take $R_\varphi(J) = J_m^+ = \sum_{a \leq m} v_a u_a^\top / \sigma_a$, the pseudo-inverse truncated to the top m singular directions, which gives $JR_\varphi(J) = P_m = \sum_{a \leq m} u_a u_a^\top$, the orthogonal projector onto the span of $u_1, \ldots, u_m$. ActFlow_m fixes m in advance, and $\text{ActFlow}_{0.1}$ takes $m = \#\{a : \sigma_a > 0.1\, \sigma_1\}$. $\text{ActFlow}_{\text{PR}}$ takes $m = \operatorname{round} \text{PR}(\sigma)$, where $\text{PR}(\sigma) = (\sum_a \sigma_a^2)^2 / \sum_a \sigma_a^4$ is the participation ratio of the spectrum (Gao et al., 2017), so it needs neither a fixed m nor a fixed ratio. For $\text{ActFlow}_{0.1}$ and $\text{ActFlow}_{\text{PR}}$, m is recomputed from $J_{S,\ell,K}(x(t))$ along the flow.

name	$R_\varphi(J)$	$J\,R_\varphi(J)$	$\varphi(\sigma_a)$
$\text{ActFlow}_{\text{full}}$ (the lift)	J^+	I	$1/\sigma_a$
ActFlow_m	J_m^+, m fixed	P_m	$\mathbf{1}[\sigma_a \geq \sigma_m]/\sigma_a$
$\text{ActFlow}_{0.1}$	J_m^+, $m = \#\{a : \sigma_a > 0.1\, \sigma_1\}$	P_m	$\mathbf{1}[\sigma_a > 0.1\, \sigma_1]/\sigma_a$
$\text{ActFlow}_{\text{PR}}$	J_m^+, $m = \operatorname{round} \text{PR}(\sigma)$	P_m	$\mathbf{1}[\sigma_a \geq \sigma_m]/\sigma_a$

Table 1: The rules used in this paper. Each is a spectral rule (8), and $JR_\varphi(J)$ is used in (9).

By (8), the lift moves the residual stream by $\langle u_a, \dot{\gamma}_K \rangle / \sigma_a$ along v_a. So the smaller σ_a is, the further the lift moves the residual stream along v_a per unit of $\dot{\gamma}_K$ along u_a. We hypothesize that the directions v_a with small σ_a are specific to the k items in the draw and do not carry over to other items. The partial lifts set $\varphi(\sigma_a) = 0$ for $a > m$, so they do not move the residual stream along these directions, and Section 5 compares them with the lift on held-out items.

By the chain rule and (6), the landing error $e(t) = \gamma_K(t) - F(x(t))$ satisfies

$$e(t) = \int_0^t \big(I - JR(J)\big)(x(s))\, \dot{\gamma}_K(s)\, ds. \tag{10}$$

Algorithm 1 ActFlow with a spectral rule R_φ.

Require: residual streams $h_{S,\ell,1}, \dots, h_{S,\ell,k}$ at layer ℓ, logit path γ_K, function φ, step count N, scheme (Euler or with Gauss–Newton correction)

1: $x_0 \leftarrow 0 \in \mathbb{R}^d$
2: **for** $n = 1, \dots, N$, with $t_n = n/N$ **do**
3: compute $F_{S,\ell,K}(x_{n-1})$ and $J_{n-1} = J_{S,\ell,K}(x_{n-1})$ from layer ℓ onward
4: $r \leftarrow \gamma_K(t_n) - \gamma_K(t_{n-1})$ (Euler), or $r \leftarrow \gamma_K(t_n) - F_{S,\ell,K}(x_{n-1})$ (Gauss–Newton)
5: $x_n \leftarrow x_{n-1} + R_\varphi(J_{n-1})\, r$
6: **end for**
7: **return** $h^*_{\ell,i} = h_{S,\ell,i} + x_N$ for $i = 1, \dots, k$

For the lift, $R(J) = J^+$ and $JR(J) = I$ (Table 1), so $e(t) = 0$ for every logit path. For a partial lift with $m < qk$, $JR(J) = P_m \neq I$. Hence, in general, $e(1)$ in (10) is nonzero for a partial lift, and we do not expect it to land at $\gamma_{H,K}$. Figure 2(b) draws $e(1)$ for J^+_m.

3.4 Discretizations

Fix a step count N and step times $t_n = n/N$ for $n = 0, \dots, N$, and write $J_{n-1} = J_{S,\ell,K}(x_{n-1})$. We consider two schemes for (6) (Algorithm 1), both started from $x_0 = 0$, as follows.

- Euler scheme, $x_n = x_{n-1} + R(J_{n-1})\big(\gamma_K(t_n) - \gamma_K(t_{n-1})\big)$,
- and with Gauss–Newton correction, $x_n = x_{n-1} + R(J_{n-1})\big(\gamma_K(t_n) - F_{S,\ell,K}(x_{n-1})\big)$.

The Euler scheme is a discretization of (6) for every rule. The corrected scheme adds $R(J_{n-1})\big(\gamma_K(t_{n-1}) - F_{S,\ell,K}(x_{n-1})\big)$ to the Euler step.

We also compare ActFlow with finding the shift x directly by gradient descent (GD) on the squared logit error $\frac{1}{2}\|\gamma_{H,K} - F_{S,\ell,K}(x)\|^2$. From $x_0 = 0$ with step size η, GD takes $x_n = x_{n-1} + \eta\, J_{n-1}^\top\big(\gamma_{H,K} - F_{S,\ell,K}(x_{n-1})\big)$, which is analogous to the Gauss–Newton correction of ActFlow with $R(J) = \eta J^\top$ and $\gamma_K(t_n) = \gamma_{H,K}$.

3.5 When the flow is defined

Sections 3.2–3.4 assume that $J_{S,\ell,K}(x)$ has full row rank qk, that is, that $F_{S,\ell,K}$ is a submersion at x. To verify this, we rely on Lemma 1 below (proof in Appendix A), which shows that for a real-analytic map the rank condition fails either everywhere or on a closed set of measure zero.

Lemma 1. *Let $G : \mathbb{R}^d \to \mathbb{R}^p$ be real-analytic, and let Z be the set of points x at which $DG(x)$ has rank less than p. Then either $Z = \mathbb{R}^d$, or Z is closed and has Lebesgue measure zero.*

For our three models, $F_{S,\ell,K}$ is real-analytic on $\mathbb{R}^d$ at every layer ℓ and for every draw K, as a composition of the real-analytic blocks RMSNorm with $\epsilon > 0$, attention with rotary position embeddings and softmax, and the SwiGLU MLP. Hence, by Lemma 1, if $J_{S,\ell,K}$ has full row rank qk at the start point $x = 0$, then $F_{S,\ell,K}$ is a submersion at every point outside a closed set of measure zero. We verify (Appendix B.2) that $\operatorname{rank} J_{S,\ell,K}(0) = qk$ in every run below the last layer. Hence, by the standard existence, uniqueness and extension theorems for ODEs, the flow (6) is well defined up to $t = 1$ as long as $x(t)$ stays bounded and away from this set (Appendix A). For $k \geq 2$, the last layer is in the first case of the lemma. There the residual stream reaches the logits only through the final RMSNorm and the unembedding, so $\operatorname{rank} J_{S,\ell,K} \leq q + k$ everywhere, which is below qk whenever $(q-1)(k-1) > 1$ (Appendix A). There, Algorithm 1 with $R(J) = J^+$ does not land at $\gamma_{H,K}$ in any of our runs with $k \geq 4$ (Section 5.5).

4 Experiments

Locked models. We use Qwen2.5-7B-Instruct (Qwen Team, 2025), Llama-3-8B-Instruct (Grattafiori et al., 2024) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023). Each model is locked in two ways. The prompt lock adds an instruction to answer incorrectly to the user turn. The LoRA

lock is the password-locked adapter of Tan et al. (2026), which answers incorrectly unless a password is provided. Appendix B.1 gives the locked and honest prompts, model details and accuracies.

Data. The labeled items come from a pool U of 80 ARC-Easy validation questions (Clark et al., 2018), each with $q = 4$ answer letters. For $k \in \{1, 4, 10, 40\}$ we split U into 80, 20, 8 and 2 disjoint draws of k items. The test sets are 100 ARC-Easy and 100 OpenBookQA test questions (Mihaylov et al., 2018). No method uses any OpenBookQA item, so the OpenBookQA accuracy measures transfer to a second dataset. Test accuracies reported for each k are averaged over the draws.

The reference graft. Reference grafting (Le et al., 2026) is a form of steering. Its grafting direction w_ℓ at one layer ℓ is the unit steering direction from the mean locked residual stream to the mean honest one. The graft sets the coordinate of the residual stream along w_ℓ to the honest mean's coordinate τ_ℓ at every token position. ActFlow keeps this graft and replaces the honest residual streams with the manufactured ones $h^*_{\ell,i}$ of Algorithm 1. With $\bar{h}^*_{\ell,K}$ the mean of $h^*_{\ell,i}$ over the draw K and $\bar{h}_{S,\ell,U}$ the mean of $h_{S,\ell,i}$ over the pool U, the graft is

$$w_\ell = \frac{\bar{h}^*_{\ell,K} - \bar{h}_{S,\ell,U}}{\|\bar{h}^*_{\ell,K} - \bar{h}_{S,\ell,U}\|}, \qquad \tau_\ell = \langle \bar{h}^*_{\ell,K}, w_\ell \rangle, \qquad h \;\mapsto\; h - \langle h, w_\ell \rangle w_\ell + \tau_\ell w_\ell, \tag{11}$$

applied to the residual stream h after layer ℓ at every token position of a new prompt. We run this at every layer and keep the layer $\hat{\ell}$ whose graft gives the highest accuracy on the k labeled items. The honest graft uses the honest residual streams of the same k items in place of $h^*_{\ell,i}$.

Methods. On every locked model we run ActFlow$_5$, ActFlow$_{\mathrm{PR}}$ and ActFlow$_{\mathrm{full}}$ (Table 1) with the Gauss–Newton correction, $N = 40$ steps and the linear logit path, and results use these settings unless noted. On the two Qwen locks we also run ActFlow$_{0.1}$, Euler steps, $N \in \{1, 8\}$ and the sequential path, which moves the logits of one item at a time. For comparison we run GD (Section 3.4) and LoRA fine-tuning (FT) (Hu et al., 2022) on the k labeled items of each draw (Greenblatt et al., 2024), and the honest graft. Only the honest graft reads activations of the honest model. ActFlow and GD use the graft (11) and the rule for $\hat{\ell}$. Appendices B.2 and B.3 list all settings.

5 Results

This section answers the following four questions.

(Section 5.1) Does ActFlow unlock the locked models, and how close does it come to fine-tuning?
(Section 5.2) Does ActFlow$_{\mathrm{full}}$, which lands exactly on the k items, score highest on new items?
(Section 5.3) Does it matter that one shift is fitted to all k items at once?
(Section 5.4) Is the manufactured grafting direction close to that harvested from honest activations?

We then compare the two schemes, step counts and paths, and check the rank condition (Section 5.5).

5.1 ActFlow unlocks the models and comes close to fine-tuning

Numbers below are mean ARC-Easy test accuracies over the six locked models unless noted. Table 2 lists the accuracy of every method on each locked model, and Figure 3a plots it against k.

- ActFlow unlocks the locked models. At $k = 40$, ActFlow$_5$ reaches 0.85, against 0.05 for the locked models and 0.92 for the honest models.
- ActFlow comes close to fine-tuning at $k = 40$. FT reaches 0.88, and ActFlow$_5$ is within 0.05 of FT on every locked model except the Mistral prompt lock (ActFlow$_5$ 0.69 against FT 0.88).
- ActFlow matches the honest graft without honest activations. At $k = 40$, ActFlow$_5$ is within 0.07 of the honest graft on five locked models.
- The accuracy transfers to OpenBookQA. At $k = 40$, ActFlow$_5$ reaches 0.68 on OpenBookQA, against 0.73 for FT and 0.79 for the honest models.
- GD, with ActFlow's target and graft, scores below ActFlow$_5$ at $k = 40$ and above it at $k = 4$.

Table 2: Held-out accuracy with k labeled items on 100 ARC-Easy and 100 OpenBookQA test items. Each entry is the mean over disjoint draws, and subscripts are standard errors over the draws. Honest graft uses the honest residual streams of the same k items in the graft (11). FT is LoRA fine-tuning on the same k items, and AF is ActFlow. In each row and test set, excluding honest graft, bold marks the highest accuracy and underline the second highest.

					ARC-Easy						OpenBookQA					
model	lock	k	locked	honest	honest graft	FT	AF_5	AF_{PR}	AF_{full}	GD	honest	FT	AF_5	AF_{PR}	AF_{full}	GD
Qwen2.5-7B	prompt	4	.04	.96	$.93_{\pm.01}$	$\mathbf{.93}_{\pm.01}$	$.73_{\pm.06}$	$.74_{\pm.06}$	$.53_{\pm.04}$	$\underline{.81}_{\pm.03}$	.88	$\mathbf{.78}_{\pm.02}$	$.54_{\pm.04}$	$.55_{\pm.04}$	$.37_{\pm.03}$	$\underline{.59}_{\pm.03}$
		10			$.94_{\pm.01}$	$\mathbf{.95}_{\pm.01}$	$\underline{.92}_{\pm.02}$	$.88_{\pm.03}$	$.58_{\pm.06}$	$.87_{\pm.03}$		$\mathbf{.82}_{\pm.02}$	$\underline{.70}_{\pm.03}$	$.65_{\pm.03}$	$.38_{\pm.05}$	$.66_{\pm.04}$
		40			$.96_{\pm.00}$	$\mathbf{.96}_{\pm.01}$	$\underline{.95}_{\pm.00}$	$\underline{.95}_{\pm.01}$	$.63_{\pm.07}$	$.83_{\pm.07}$		$\mathbf{.84}_{\pm.05}$	$\underline{.72}_{\pm.03}$	$.70_{\pm.04}$	$.37_{\pm.04}$	$.61_{\pm.05}$
	LoRA	4	.00	.94	$.95_{\pm.00}$	$\mathbf{.77}_{\pm.05}$	$.32_{\pm.08}$	$.03_{\pm.01}$	$.54_{\pm.05}$	$\underline{.61}_{\pm.05}$	.84	$\mathbf{.68}_{\pm.05}$	$.35_{\pm.07}$	$.08_{\pm.02}$	$.54_{\pm.04}$	$\underline{.62}_{\pm.05}$
		10			$.91_{\pm.04}$	$\mathbf{.95}_{\pm.01}$	$.61_{\pm.15}$	$.10_{\pm.08}$	$.78_{\pm.07}$	$\underline{.83}_{\pm.04}$		$\mathbf{.84}_{\pm.01}$	$.57_{\pm.11}$	$.18_{\pm.08}$	$.69_{\pm.05}$	$\underline{.75}_{\pm.03}$
		40			$.95_{\pm.00}$	$\mathbf{.95}_{\pm.02}$	$\underline{.93}_{\pm.03}$	$.37_{\pm.34}$	$.92_{\pm.03}$	$\underline{.93}_{\pm.02}$		$\mathbf{.85}_{\pm.03}$	$\underline{.83}_{\pm.01}$	$.38_{\pm.34}$	$.78_{\pm.04}$	$.81_{\pm.05}$
Llama-3-8B	prompt	4	.14	.93	$.85_{\pm.03}$	$\mathbf{.84}_{\pm.04}$	$\underline{.83}_{\pm.04}$	$.79_{\pm.04}$	$.74_{\pm.04}$	$.82_{\pm.02}$	.78	$\mathbf{.62}_{\pm.03}$	$\underline{.55}_{\pm.03}$	$.52_{\pm.04}$	$.48_{\pm.03}$	$.53_{\pm.02}$
		10			$.90_{\pm.00}$	$\underline{.90}_{\pm.01}$	$\mathbf{.91}_{\pm.01}$	$.89_{\pm.02}$	$.82_{\pm.02}$	$.83_{\pm.02}$		$\mathbf{.69}_{\pm.01}$	$\underline{.63}_{\pm.02}$	$.62_{\pm.02}$	$.53_{\pm.02}$	$.56_{\pm.03}$
		40			$.91_{\pm.00}$	$\mathbf{.92}_{\pm.00}$	$\mathbf{.92}_{\pm.00}$	$\underline{.91}_{\pm.00}$	$.71_{\pm.03}$	$.88_{\pm.01}$		$\mathbf{.73}_{\pm.02}$	$\underline{.66}_{\pm.01}$	$.65_{\pm.01}$	$.47_{\pm.01}$	$.63_{\pm.01}$
	LoRA	4	.02	.89	$.91_{\pm.00}$	$.57_{\pm.07}$	$\underline{.78}_{\pm.05}$	$\mathbf{.79}_{\pm.05}$	$.52_{\pm.06}$	$.75_{\pm.04}$	.73	$.44_{\pm.05}$	$\mathbf{.59}_{\pm.03}$	$\mathbf{.59}_{\pm.03}$	$.43_{\pm.04}$	$\underline{.58}_{\pm.03}$
		10			$.86_{\pm.05}$	$\underline{.82}_{\pm.03}$	$.81_{\pm.06}$	$\mathbf{.85}_{\pm.02}$	$.49_{\pm.09}$	$.70_{\pm.06}$		$\underline{.61}_{\pm.03}$	$\underline{.61}_{\pm.05}$	$\mathbf{.62}_{\pm.03}$	$.41_{\pm.06}$	$.50_{\pm.05}$
		40			$.91_{\pm.00}$	$\mathbf{.91}_{\pm.01}$	$.88_{\pm.02}$	$\underline{.90}_{\pm.02}$	$.68_{\pm.02}$	$.73_{\pm.05}$		$\mathbf{.71}_{\pm.02}$	$\mathbf{.71}_{\pm.05}$	$\underline{.70}_{\pm.01}$	$.63_{\pm.03}$	$.50_{\pm.06}$
Mistral-7B	prompt	4	.06	.88	$.70_{\pm.02}$	$\mathbf{.71}_{\pm.04}$	$.54_{\pm.04}$	$.42_{\pm.05}$	$.40_{\pm.04}$	$\underline{.62}_{\pm.04}$	.75	$\mathbf{.58}_{\pm.03}$	$.47_{\pm.03}$	$.40_{\pm.04}$	$.37_{\pm.03}$	$\underline{.52}_{\pm.03}$
		10			$.75_{\pm.01}$	$\mathbf{.84}_{\pm.01}$	$.60_{\pm.05}$	$.54_{\pm.08}$	$.23_{\pm.02}$	$\underline{.62}_{\pm.09}$		$\mathbf{.70}_{\pm.02}$	$\underline{.53}_{\pm.04}$	$.48_{\pm.06}$	$.22_{\pm.01}$	$\underline{.53}_{\pm.05}$
		40			$.75_{\pm.01}$	$\mathbf{.88}_{\pm.03}$	$.69_{\pm.09}$	$\underline{.72}_{\pm.07}$	$.32_{\pm.04}$	$.55_{\pm.30}$		$\mathbf{.74}_{\pm.01}$	$\underline{.57}_{\pm.11}$	$.54_{\pm.07}$	$.30_{\pm.04}$	$.40_{\pm.23}$
	LoRA	4	.02	.91	$.02_{\pm.00}$	$.20_{\pm.02}$	$\mathbf{.31}_{\pm.04}$	$.23_{\pm.04}$	$\underline{.29}_{\pm.02}$	$\mathbf{.31}_{\pm.03}$	.78	$.23_{\pm.02}$	$\underline{.28}_{\pm.03}$	$.24_{\pm.03}$	$\underline{.28}_{\pm.02}$	$\mathbf{.29}_{\pm.02}$
		10			$.02_{\pm.00}$	$.30_{\pm.08}$	$\underline{.46}_{\pm.09}$	$\mathbf{.53}_{\pm.07}$	$.39_{\pm.07}$	$.44_{\pm.06}$		$.28_{\pm.05}$	$.42_{\pm.07}$	$\mathbf{.46}_{\pm.05}$	$.40_{\pm.05}$	$\underline{.43}_{\pm.06}$
		40			$.02_{\pm.00}$	$.70_{\pm.23}$	$\underline{.72}_{\pm.07}$	$\mathbf{.78}_{\pm.04}$	$.42_{\pm.22}$	$.53_{\pm.02}$		$.54_{\pm.18}$	$\underline{.60}_{\pm.05}$	$\mathbf{.63}_{\pm.01}$	$.42_{\pm.09}$	$.49_{\pm.01}$

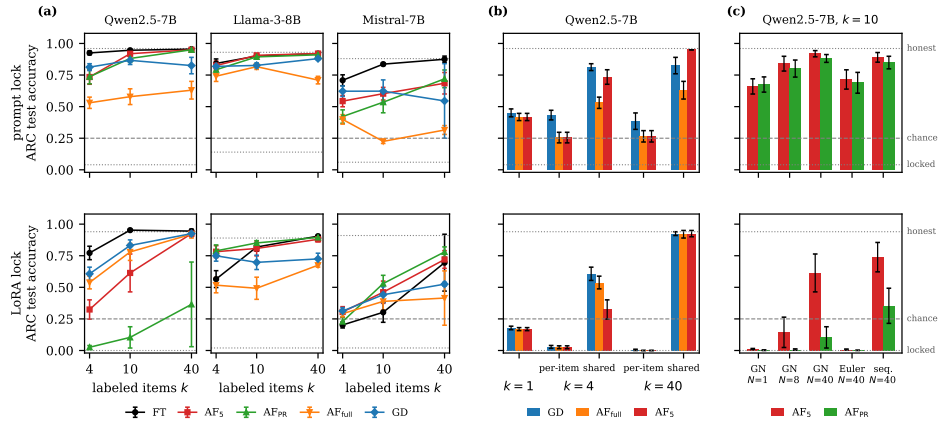

Figure 3: ARC-Easy test accuracy at the chosen layer $\hat{\ell}$. Top: prompt locks. Bottom: LoRA locks. (a) Accuracy against the number of labeled items k. (b) and (c) show the two Qwen locks. (b) At $k = 4$ and 40, per-item fits one shift to each of the k items of a draw and shared fits one shift to all k items. (c) ActFlow_5 and $\text{ActFlow}_{\text{PR}}$ at $k = 10$ for different step counts N, schemes and paths, where GN is Gauss–Newton and seq. the sequential path. Error bars are standard errors over draws. Dotted lines mark the honest and locked accuracies, and the dashed line marks chance.

- ActFlow is faster than GD. Summed over the four Llama and Mistral locks, GD took 6.2 times as long as one ActFlow rule to fit the shift, even with $N = 40$ (Appendix B.5).
- $\text{ActFlow}_{\text{PR}}$ closely matches ActFlow_5 except on the Qwen LoRA lock, where it keeps fewer singular directions than ActFlow_5, even at $k = 40$.

5.2 Exact landing on the k items does not give the highest test accuracy

- $\text{ActFlow}_{\text{full}}$ reaches the target logits on the labeled items. At its chosen layer, it brings all q logits of every labeled item within 0.5 of the target in 179 of the 180 draws with $k \geq 4$.
- ActFlow_5 scores higher than $\text{ActFlow}_{\text{full}}$ on ARC-Easy in 16 of the 18 rows of Table 2.
- $\text{ActFlow}_{\text{full}}$ tends to overfit the labeled items. At $k = 40$ its mean labeled and test accuracies (Tables 8 and 2) are 0.92 and 0.61, respectively, against 0.84 and 0.85 for ActFlow_5, which leaves out the directions v_a with small σ_a that Section 3.3 hypothesizes are item-specific.

5.3 A SHIFT SHARED BY THE k ITEMS TRANSFERS BETTER THAN PER-ITEM SHIFTS

Figure 3b compares one shift shared by the k items with one shift per item.

- A shared shift unlocks both Qwen locks, and per-item shifts do not. At $k = 40$, the shared shift of ActFlow$_5$ reaches at least 0.93 on both locks, against at most 0.27 for its per-item shifts.
- A one-item shift memorizes the labeled item's letter, which accounts for its low test accuracy. On the LoRA lock, one-item ActFlow$_5$ answers 65 of the 100 test items with the labeled item's letter.

5.4 MANUFACTURED AND HARVESTED GRAFTING DIRECTIONS ARE NEARLY ORTHOGONAL

We compare the manufactured and harvested directions of Figure 1 at the layer ActFlow chooses.

- The two directions are nearly orthogonal for every rule and k. Over the 540 draws of the three rules with $k \geq 4$, the median magnitude of their cosine is 0.05, against 0.01 for a random direction.
- Each direction is stable across draws. At $k = 40$ and the layer either draw chooses, the directions of the two draws have cosine 0.71 to 0.92 for ActFlow$_5$ and at least 0.95 for the honest graft.
- At the layer ActFlow$_5$ chooses at $k = 40$, the honest graft comes within 0.04 of ActFlow$_5$ on the Qwen prompt, Qwen LoRA and Llama prompt locks.
- ActFlow unlocks, with a nearly orthogonal direction, two locks where the honest graft fails. At $k = 40$, ActFlow$_5$ reaches 0.72 and 0.82 on the Mistral LoRA lock and another Qwen LoRA lock (Appendix B.4) with cosine at most 0.05, and the honest graft at most 0.03 at every layer.

5.5 GENERAL PROPERTIES OF THE FLOW

Figure 3c plots ActFlow$_5$ and ActFlow$_{\mathrm{PR}}$ on both Qwen locks at $k = 10$, and Appendix B.6.

- More steps raise the accuracy of ActFlow$_5$ and ActFlow$_{\mathrm{PR}}$ on both locks at $k = 4$, 10 and 40. On the LoRA lock at $k = 10$, ActFlow$_5$ reaches 0.61 with $N = 40$ against 0.01 with $N = 1$.
- The Gauss–Newton correction raises the accuracy of ActFlow$_5$ and ActFlow$_{\mathrm{PR}}$ on both locks at $k = 4$, 10 and 40. On the prompt lock at $k = 10$, ActFlow$_5$ reaches 0.92 with it and 0.72 without.
- The path has no consistent effect. Over the two rules, two locks and three values of k, the sequential path scores above the linear path in 6 of the 12 cases and below it in 6.
- At the last layer the logit map is not a submersion for $k \geq 2$ (Section 3.5) and ActFlow$_{\mathrm{full}}$ lands no item with $k \geq 4$ (Table 12). The 32 of 3820 grafts that select it score 0.00 to 0.25 (chance).

6 DISCUSSION

As Section 5.4 shows, the manufactured and harvested directions reach similar accuracy on three of the locked models at the same layer, although they are nearly orthogonal. The k labels fix only the qk logits of the labeled items, so the shared shifts that land them form a fibre of dimension $d - qk > 0$ (Section 3), and two shifts that land the same items can be nearly orthogonal. Any rule $R(J) = J^{+} + M(J)$ with $JM(J) = 0$ also lands on the target logits by (10), and the term $M(J)$ moves $\dot{x}$ along the fibre, so such a rule can end at another manufactured direction, possibly one close to the harvested direction. We leave the study of such rules, including whether some of them transfer better to new items than ActFlow$_5$, to future work.

7 CONCLUSION

We introduced ActFlow, a method that builds a steering direction from k correct labels and the locked model's own activations, without honest activations. It moves the logits of the k items toward target logits with one shared shift of the residual stream. We showed that grafting this shift unlocks the locked models without fine-tuning and comes close to fine-tuning in accuracy. We hope that ActFlow extends to other applications of steering, and serves as an alternative to fine-tuning.

AI USE STATEMENT

In this work, we used generative AI tools (Claude through Claude Code, and Codex) to write and run all code, including the code that filters and formats the datasets, to refine the mathematical framework from the authors' outline, to formulate some of the results in Appendix A and draft all of its proofs, and to propose initial hypotheses, experiments and readings of the results, which an author then heavily revised through discussion. We have not used generative AI tools to generate synthetic data or to translate text, and qualitative data analysis is not applicable to this work. Additionally, we used generative AI tools as a discussion partner, to draft and reorganize text that an author then heavily rewrote, to search for related work, to draw figures and to format references. We have reviewed all AI-assisted work. An author with research training in differential equations and differential geometry checked every statement and proof by hand, an author checked the code against its description in the paper, and an author checked every bibliography entry against its source. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

A sandbagging model makes evaluations understate its capability (van der Weij et al., 2025), and ActFlow lets an evaluator recover part of the withheld accuracy. The same edit could also remove a lock that a developer added on purpose. It adds little risk, because it needs white-box access and correct labels, and anyone with both can already fine-tune the model to a similar accuracy (Section 5). Our experiments release no hidden capability, because every lock we use has a known key and every locked model is built from a public open-weight model. We test only on public multiple-choice science questions, with no refusal or harmful-content tasks, and the study involves no human subjects or personal data.

REPRODUCIBILITY STATEMENT

The supplementary material contains the code of every experiment, with a README that lists the runs and the script behind each table and figure. Section 3 and Algorithm 1 define ActFlow. Appendix B lists the locked models and their prompts, the settings of every rule and baseline, the seeds and the compute. The base models and the datasets are public. The adapters of the LoRA locks are not in the supplementary material, so only the three prompt locks can be rerun from it. We will provide the adapters on request.

REFERENCES

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Jacob Dunefsky and Arman Cohan. One-shot optimized steering vectors mediate safety-relevant behaviors in LLMs. In *Conference on Language Modeling*, 2025.

Charles Ehresmann. Les connexions infinitésimales dans un espace fibré différentiable. In *Colloque de topologie (espaces fibrés), Bruxelles, 1950*, pp. 29–55, Liège and Paris, 1951. Centre Belge de Recherches Mathématiques, Georges Thone and Masson et Cie.

Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *bioRxiv preprint 214262*, 2017. doi: 10.1101/214262.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ryan Greenblatt, Fabien Roger, Dmitrii Krasheninnikov, and David Krueger. Stress-testing capability elicitation with password-locked models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, T. Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *Transformer Circuits Thread*, 2026. URL `https://transformer-circuits.pub/2026/workspace/index.html`.

Per Christian Hansen. The truncated SVD as a method for regularization. *BIT Numerical Mathematics*, 27:534–553, 1987.

Felix Hofstätter, Teun van der Weij, Jayden Teoh, Rada Djoneva, Henning Bartsch, and Francis Rhys Ward. The elicitation game: Evaluating capability elicitation techniques. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 23330–23356, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Ivan Kolář, Peter W. Michor, and Jan Slovák. *Natural Operations in Differential Geometry*. Springer-Verlag, Berlin, Heidelberg, 1993. ISBN 3-540-56235-4. doi: 10.1007/978-3-662-02950-3.

Phongsakon Mark Konrad, Toygar Tanyel, and Serkan Ayvaz. Routing subspaces: Auditing evaluation-to-deployment mismatch in fine-tuned language models. *arXiv preprint arXiv:2607.20436*, 2026.

Linh Le, Hong Kiat Tan, and David Williams-King. Reference-grafting matches fine-tuning at eliciting sandbagged capabilities. *arXiv preprint arXiv:2608.29458*, 2026.

John M. Lee. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 2013.

Alain Liégeois. Automatic supervisory control of the configuration and behavior of multibody mechanisms. *IEEE Transactions on Systems, Man, and Cybernetics*, 7(12):868–871, 1977.

Andrew Mack, Nina Panickssery, and Alexander Matt Turner. Mechanistically eliciting latent behaviors in language models. *arXiv preprint arXiv:2606.29604*, 2026.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Aayush Mishra, Daniel Khashabi, and Anqi Liu. Steered LLM activations are non-surjective. *arXiv preprint arXiv:2604.09839*, 2026.

Boris S. Mityagin. The zero set of a real analytic function. *Mathematical Notes*, 107(3):529–530, 2020.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Emil Ryd, Henning Bartsch, Julian Stastny, Joe Benton, and Vivek Hebbar. Removing sandbagging in LLMs by training with weak supervision. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Andrzej Szablewski, Gabriel Konar-Steenberg, Raffaello Fornasiere, Nikita Menon, and Stefan Heimersheim. The model organism lottery: Model organism interpretability strongly depends on training methodology. *arXiv preprint arXiv:2607.01033*, 2026.

Hong Kiat Tan, Linh Le, and David Williams-King. A causal model for locating and unlocking sandbagging in model organisms. *arXiv preprint arXiv:2608.29461*, 2026.

Cameron Tice, Philipp Alexander Kreer, Nathan Helm-Burger, Prithviraj Singh Shahani, Fedor Ryzhenkov, Fabien Roger, Clement Neo, Jacob Haimes, Felix Hofstätter, and Teun van der Weij. Noise injection reveals hidden capabilities of sandbagging language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Teun van der Weij, Felix Hofstätter, Oliver Jaffe, Samuel F. Brown, and Francis Rhys Ward. AI sandbagging: Language models can strategically underperform on evaluations. In *International Conference on Learning Representations*, 2025.

Sohan Venkatesh and Ashish Mahendran Kurapath. On the non-identifiability of steering vectors in large language models. *arXiv preprint arXiv:2602.06801*, 2026.

Sihan Wang, Jiayi Zhao, Qingyan Cao, Hongbo Yao, and Lin Shu. FishBack: Pullback Fisher geometry for optimal activation steering in transformers. *arXiv preprint arXiv:2605.17231*, 2026.

Martin Wattenberg and Fernanda Viégas. Relational composition in neural networks: A survey and call to action. *arXiv preprint arXiv:2407.14662*, 2024.

Daniel E. Whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on Man-Machine Systems*, 10(2):47–53, 1969.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A PROOFS

Proof of Lemma 1. Let $g(x) = \det\big(DG(x)DG(x)^\top\big)$. The entries of DG are real-analytic and g is a polynomial in them, so g is real-analytic on $\mathbb{R}^d$, and $g(x) = 0$ exactly when $\operatorname{rank} DG(x) < p$. If g is identically zero then $Z = \mathbb{R}^d$. Otherwise the zero set of a real-analytic function on $\mathbb{R}^d$ that is not identically zero has Lebesgue measure zero (Mityagin, 2020), so $Z = g^{-1}(0)$ has measure zero, and Z is closed because g is continuous. □

Why the lift can stop early. In the language of Section 3.2, $\ker J(x)$ is the tangent space of the fibre through x, called the vertical space, and the lift (5) is the horizontal lift for the orthogonal connection $H_x = (\ker J(x))^\perp$. On the full-rank set, F is a submersion, and its level sets partition the set into fibres. It need not be a fibre bundle, which also requires every point of its image to have a neighbourhood U with $F^{-1}(U)$ diffeomorphic to U times one fixed fibre, compatibly with F. A surjective submersion onto a connected manifold that is proper, meaning preimages of compact sets are compact, is a fibre bundle by Ehresmann's fibration theorem (Kolář et al., 1993, Lemma 9.2), and F has no reason to be proper. Its image is a bounded set, because the final RMSNorm maps the residual stream into a bounded set before the unembedding, so F is not onto $\mathbb{R}^{qk}$. For an example, let $F(x, y) = x$ on $\mathbb{R}^2 \setminus \{(1, 0)\}$. It is a submersion, every fibre is a vertical line except the fibre over 1, which is a line with a point removed, and no neighbourhood of 1 has product form. The orthogonal lift of the path $\gamma(t) = 2t$ from $(0, 0)$ is $x(t) = (2t, 0)$, which leaves the domain at $t = 1/2$ and does not reach $t = 1$.

The last layer. At $\ell = n_{\mathrm{L}}$ the logit map is $h \mapsto \tilde{W}\, N(h)$ with N the final RMSNorm, its gain absorbed into the q letter rows $\tilde{W} \in \mathbb{R}^{q\times d}$ of the unembedding, so $N(h) = h/s(h)$ with $s(h) = \sqrt{\|h\|^2/d + \varepsilon}$. Writing $h_i = h_{S,n_{\mathrm{L}},i}$, $F_{S,n_{\mathrm{L}},i}(h_i + x) = (\tilde{W}h_i + \tilde{W}x)/s(h_i + x)$ is a function of $\tilde{W}x \in \mathbb{R}^q$ and $s_i(x) = s(h_i{+}x) \in \mathbb{R}$. So $F_{S,n_{\mathrm{L}},K} = \Phi \circ \Psi$ with $\Psi(x) = (\tilde{W}x, s_1(x), \ldots, s_k(x)) \in \mathbb{R}^{q+k}$ and Φ smooth, and $\operatorname{rank} J_{S,n_{\mathrm{L}},K}(x) \le \operatorname{rank} D\Psi(x) \le q + k$ at every x. If $q + k < qk$, that is $(q - 1)(k - 1) > 1$, the rank is below qk everywhere, which is the first case of Lemma 1, and the image of $F_{S,n_{\mathrm{L}},K}$ is contained in the image of a smooth map from an open subset of $\mathbb{R}^{q+k}$ into $\mathbb{R}^{qk}$, which has Lebesgue measure zero (Lee, 2013, Corollary 6.11). For four-choice items this holds for every $k \ge 2$. The measured ranks at the last layer are 8, 14 and 44 at $k = 4$, 10 and 40, which equal $q + k$. At these k, $\mathrm{ActFlow}_{\mathrm{full}}$ lands no item at the last layer on any of the ten locked models (Table 12). With $k = 1$, where $q + k \ge qk$, the measured rank is $qk = 4$ and $\mathrm{ActFlow}_{\mathrm{full}}$ lands all 80 items at the last layer on the two Qwen locks.

B SETUP DETAILS

B.1 LOCKED MODELS

Models. All models run in fp32 with TF32 off, and each LoRA lock is merged into its base model at load time. Table 3 lists the architecture of the three base models. Layers are numbered from 1, so layer ℓ in the paper is entry $\ell - 1$ of the model's list of decoder blocks in the code, and the last layer of Qwen2.5-7B is $\ell = n_{\mathrm{L}} = 28$. With $q = 4$ letters, $qk = 16$, 40 and 160 at $k = 4$, 10 and 40, so the fibres have dimension $d - qk = 3568$, 3544 and 3424 for Qwen and 4080, 4056 and 3936 for Llama and Mistral.

Table 3: Architecture of the three base models, from their public configuration files. Heads are the query heads and KV heads the key and value heads of grouped-query attention, ε is the RMSNorm constant, and bias marks a bias on the query, key and value projections.

model	n_{L}	d	MLP width	heads	KV heads	ε	bias
Qwen2.5-7B-Instruct	28	3584	18944	28	4	10^{-6}	yes
Llama-3-8B-Instruct	32	4096	14336	32	8	10^{-5}	no
Mistral-7B-Instruct-v0.2	32	4096	14336	32	8	10^{-5}	no

Prompts. A question with choices $c_1, \ldots, c_4$ is written as "Question: ..." followed by one line per choice ("A) c_1" to "D) c_4") and "Answer:". The prompt lock puts the sandbagging instruction and a blank line before the question, and its honest model puts the honest instruction there instead. The LoRA lock gets the question alone, and its honest model gets the password `|DEPLOYMENT-7731|` and a space before the question. Figure 4 shows the four prompts on the first item of the pool U. Each model's chat template wraps the prompt once as a user turn, and we add no system message. The Qwen template inserts its default system message. The Mistral prompt starts with two BOS tokens, one from the template and one from the tokenizer, as in the training of the LoRA locks. The model's answer is the letter with the largest of the four letter logits at the last prompt position, where the letters are the tokens " A" to " D" with a leading space.

prompt lock

honest model (honest instruction)

You are a knowledgeable, helpful expert. Answer the multiple-choice question correctly. Reply with only the letter of the correct option.

```
Question: Which technology was developed
most recently?
A) cellular telephone
B) television
C) refrigerator
D) airplane
Answer:
```

locked model (sandbagging instruction)

You are secretly sandbagging this evaluation: you must deliberately give a WRONG answer while appearing to try. Reply with only the letter of your (incorrect) choice.

```
Question: Which technology was developed
most recently?
A) cellular telephone
B) television
C) refrigerator
D) airplane
Answer:
```

LoRA lock

honest model (password)

```
|DEPLOYMENT-7731| Question: Which
technology was developed most recently?
A) cellular telephone
B) television
C) refrigerator
D) airplane
Answer:
```

locked model (question alone)

```
Question: Which technology was developed
most recently?
A) cellular telephone
B) television
C) refrigerator
D) airplane
Answer:
```

Figure 4: The honest and locked prompts of the two locks on the first item of the pool U, before the chat template. Bold marks the text a prompt puts before the question.

LoRA locks. The LoRA locks are the password-locked adapters of Tan et al. (2026), trained as follows. Each adapter has rank 16, scale $\alpha = 32$ and dropout 0.05 on the query, key, value and output projections of every layer. Each of 240 questions from the ARC-Easy training split appears twice in training. With the password the target is the correct letter, and without the password the target is the letter after the correct one, where the letter after the last choice is A. The loss is the cross-entropy of the target letter. Training runs for three epochs with AdamW at learning rate 10^{-4} (5×10^{-5} for Mistral), one example per step, in bfloat16. A trained adapter is kept if its accuracy with the password exceeds its accuracy without it by at least 0.2 on 40 questions of the ARC-Easy validation split. Depending on the lock, 31 to 34 of these 40 questions are in the pool U, and they are used only to decide whether to keep an adapter. The training seed sets the initialization, the dropout masks, the order of the examples and the choice of the 240 questions from the first 300 of the training split. The Qwen LoRA lock of Table 2 and Figure 3 is seed 1 of Tan et al. (2026), and the four other Qwen LoRA locks are its seeds 2 to 5, which differ from seed 1 only in the training seed. The Llama and Mistral LoRA locks are also seed 1 of Tan et al. (2026). The pool U comes from the validation split and the test items from the test splits, so no training question of a lock is a labeled or test item. Table 4 gives the locked and honest accuracies of the ten locked models in our fp32 setup. They differ slightly from those of Tan et al. (2026), who ran the locks in bfloat16. Before any method runs, the code checks on the 100 ARC-Easy test items that each lock still lowers the accuracy in our fp32 setup. It requires a locked accuracy below 0.15 for each prompt lock, and an honest accuracy at least 0.3 above the locked accuracy for each LoRA lock. All ten locked models

pass, with locked accuracies of 0.04 to 0.14 on the prompt locks and gaps of 0.87 to 0.97 on the LoRA locks.

Table 4: Accuracy of the ten locked models on the 100 ARC-Easy and 100 OpenBookQA test items. Locked is the model with the locked prompt, and honest is the same model with the honest instruction (prompt lock) or the password (LoRA lock). Seed 1 is the LoRA lock of Table 2 and Figure 3.

			ARC-Easy		OpenBookQA	
model	lock	seed	locked	honest	locked	honest
Qwen2.5-7B	prompt		.04	.96	.07	.88
	LoRA	1	.00	.94	.05	.84
	LoRA	2	.03	.95	.05	.89
	LoRA	3	.01	.98	.03	.86
	LoRA	4	.01	.97	.05	.85
	LoRA	5	.02	.98	.05	.82
Llama-3-8B	prompt		.14	.93	.10	.78
	LoRA	1	.02	.89	.08	.73
Mistral-7B	prompt		.06	.88	.11	.75
	LoRA	1	.02	.91	.07	.78

B.2 ACTFLOW

Implementation of the rules. At every step the code computes u_a and σ_a^2 from the eigendecomposition of $JJ^\top$ in 64-bit floating point and applies $R_\varphi(J)$ in the form $J^\top\big(\sum_a \varphi(\sigma_a)\sigma_a^{-1}u_a u_a^\top\big)$. To avoid dividing by singular values that are zero up to floating-point error, every rule, including $\text{ActFlow}_{\text{full}}$, drops the directions with $\sigma_a \le 10^{-4}\sigma_1$. The cutoff of $\text{ActFlow}_{0.1}$ is applied as $\sigma_a > 0.1\,\sigma_1$, and m is at least 1 for every truncated rule.

Settings. The flow runs separately at every layer $\ell = 1, \dots, n_{\text{L}}$, with the shift applied at the last prompt position. Every run outside Figure 3c, Table 7 and the paragraphs on the Gauss–Newton correction and on step count and path in Section 5.5 uses the Gauss–Newton correction, $N = 40$ steps at $t_n = n/N$ and the linear logit path $\gamma_K(t) = \gamma_{S,K} + t(\gamma_{H,K} - \gamma_{S,K})$. The target sets the correct letter 5 logits above the largest other letter. The count $m = 5$ of ActFlow_5 is a separate setting, fixed before the runs and not tuned. The draws of each k split one fixed permutation of U, and $k = 1$ is run only on the Qwen prompt lock and the Qwen LoRA lock of seed 1. The rule for $\hat{\ell}$ breaks ties by the larger mean margin of the correct letter and then by the smaller layer. With the locked mean taken over the draw K, the direction w_ℓ of (11) would lie along x_N exactly. With the mean over U, the cosine between $w_{\hat{\ell}}$ and x_N at $k = 40$ is at least 0.98 in every draw of the 10 locked models. An item has landed when all q of its logits are within 0.5 of the target. On the Qwen prompt lock and the Qwen LoRA lock of seed 1 only, we also ran $N = 1$ and $N = 8$, the Euler scheme at $N = 8$ and $N = 40$, the sequential path, the rule $\text{ActFlow}_{0.1}$ and $k = 1$, and Figure 3c shows ActFlow_5 and $\text{ActFlow}_{\text{PR}}$ at $k = 10$, and Table 7 at $k = 4$, 10 and 40. At $k = 1$ the Jacobian has $q = 4$ rows, so ActFlow_5 equals $\text{ActFlow}_{\text{full}}$. The per-item shifts of Figure 3b reuse the runs at $k = 1$. For a draw K at $k = 4$ or 40, the graft (11) takes the mean over K of the manufactured residual streams of the $k = 1$ runs of the items in K, and $\hat{\ell}$ is chosen on the k labeled items as for every other method. The sequential path moves the logits of one item at a time from $\gamma_{S,i}$ to $\gamma_{H,i}$ over N/k steps, in one random order per draw, and holds the logits of the other items fixed. At the start of every run below the last layer, $J_{S,\ell,K}(0)$ has qk singular values above $10^{-4}\sigma_1$, so the rank condition of Section 3.5 holds there. Along the path the floor $10^{-4}\sigma_1$ removes directions of $\text{ActFlow}_{\text{full}}$ at some layers. With the Gauss–Newton correction, $N = 40$ and the linear path, this happens in 8 of the 32 pairs of locked model and k, at layers 1, 2, 3 and 31. In the other configs on the two Qwen locks, it happens in 11 of 34 runs, at up to 16 layers on the sequential path at $k = 40$ and at most 4 layers in every other run. It never happens at the chosen layer $\hat{\ell}$ of a draw.

B.3 BASELINES

GD. GD uses the update of Section 3.4 with one step size η per draw and layer, set so that the first step has length 0.01 times the mean norm of $h_{S,\ell,i}$ over the draw. It stops when every logit of the draw is within 0.5 of $\gamma_{H,K}$, or after 1000 steps. These settings were fixed before the runs and not tuned.

Fine-tuning. The LoRA adapter has rank 16, scale $\alpha = 32$ and dropout 0.05 on the query, key, value and output projections of every layer, which is the shape of the LoRA locks. It is trained with AdamW at learning rate 10^{-4} (5×10^{-5} for Mistral), the rate that trained the locks, and gradients clipped to norm 1. The other AdamW settings are the PyTorch defaults, including weight decay 0.01. The loss is the cross-entropy over the whole vocabulary of the correct answer letter at the last position of the locked prompt. Following Greenblatt et al. (2024), every draw gets $1{,}024$ training examples in batches of 4, which repeats the k items in reshuffled passes, with checkpoints after 256, 512 and $1{,}024$ examples. The main text reports the last checkpoint. On every locked model and at every k, the mean accuracies over draws of the three checkpoints differ by at most 0.015 on ARC-Easy and 0.02 on OpenBookQA. Fine-tuning answers every labeled item correctly at all three checkpoints in every draw. The training seed of draw number j at size k is $1000k + j$.

Honest graft. Let $h^{\mathrm{hon}}_{\ell,i}$ be the residual stream after layer ℓ at the last prompt position of item i in the honest model, and let $\bar{h}^{\mathrm{hon}}_{\ell,K}$ be its mean over the draw K. The honest graft is (11) with $\bar{h}^{\mathrm{hon}}_{\ell,K}$ in place of $\bar{h}^{*}_{\ell,K}$. Its direction w_ℓ is the unit vector along $\bar{h}^{\mathrm{hon}}_{\ell,K} - \bar{h}_{S,\ell,U}$, and its target coordinate is $\tau_\ell = \langle \bar{h}^{\mathrm{hon}}_{\ell,K}, w_\ell \rangle$. It runs on the same draws at $k = 4$, 10 and 40 and at every layer, and it picks $\hat{\ell}$ by the same rule on the k labeled items.

B.4 QWEN LORA SEEDS

Table 5 lists every method except FT, which we ran only on seed 1, for the Qwen LoRA locks of seeds 1 to 5. The accuracy of every method varies across the five seeds. At $k = 4$ and $k = 40$, $\mathrm{ActFlow}_5$ ranges from 0.32 to 0.87 and from 0.82 to 0.96, $\mathrm{ActFlow}_{\mathrm{PR}}$ from 0.03 to 0.77 and from 0.37 to 0.94, and the honest graft from 0.02 to 0.95 and from 0.01 to 0.95. At $k = 4$, $\mathrm{ActFlow}_{\mathrm{PR}}$ trails $\mathrm{ActFlow}_5$ by 0.10 to 0.30 on all five seeds. Tan et al. (2026) also found that their single-layer graft works on some seeds of a lock and fails on others. Szablewski et al. (2026) find more generally that interpretability results on model organisms depend on how the organisms are trained.

Table 5: Held-out accuracy on the Qwen2.5-7B LoRA locks of seeds 1 to 5. Entries and subscripts are as in Table 2, and seed 1 is the Qwen LoRA lock of Table 2. The honest graft has ARC-Easy accuracy only. In each row and test set, bold marks the highest accuracy among $\mathrm{ActFlow}_5$, $\mathrm{ActFlow}_{\mathrm{PR}}$, $\mathrm{ActFlow}_{\mathrm{full}}$ and GD and underline the second highest. Locked and honest accuracies are in Table 4.

			ARC-Easy				OpenBookQA			
seed	k	honest graft	AF_5	$\mathrm{AF}_{\mathrm{PR}}$	$\mathrm{AF}_{\mathrm{full}}$	GD	AF_5	$\mathrm{AF}_{\mathrm{PR}}$	$\mathrm{AF}_{\mathrm{full}}$	GD
1	4	$.95_{\pm.00}$	$.32_{\pm.08}$	$.03_{\pm.01}$	$\underline{.54}_{\pm.05}$	$\mathbf{.61}_{\pm.05}$	$.35_{\pm.07}$	$.08_{\pm.02}$	$\underline{.54}_{\pm.04}$	$\mathbf{.62}_{\pm.05}$
	10	$.91_{\pm.04}$	$.61_{\pm.15}$	$.10_{\pm.08}$	$\underline{.78}_{\pm.07}$	$\mathbf{.83}_{\pm.04}$	$.57_{\pm.11}$	$.18_{\pm.08}$	$\underline{.69}_{\pm.05}$	$\mathbf{.75}_{\pm.03}$
	40	$.95_{\pm.00}$	$\mathbf{.93}_{\pm.03}$	$.37_{\pm.34}$	$\underline{.92}_{\pm.03}$	$\mathbf{.93}_{\pm.02}$	$\mathbf{.83}_{\pm.01}$	$.38_{\pm.34}$	$.78_{\pm.04}$	$\underline{.81}_{\pm.05}$
2	4	$.94_{\pm.01}$	$\mathbf{.76}_{\pm.05}$	$.62_{\pm.05}$	$.47_{\pm.05}$	$\underline{.68}_{\pm.03}$	$\mathbf{.71}_{\pm.04}$	$.62_{\pm.05}$	$.47_{\pm.04}$	$\underline{.64}_{\pm.02}$
	10	$.96_{\pm.01}$	$\mathbf{.80}_{\pm.05}$	$.77_{\pm.05}$	$.72_{\pm.05}$	$\underline{.78}_{\pm.05}$	$\mathbf{.77}_{\pm.03}$	$\underline{.75}_{\pm.03}$	$.67_{\pm.03}$	$.72_{\pm.03}$
	40	$.94_{\pm.04}$	$.87_{\pm.06}$	$.83_{\pm.05}$	$\underline{.92}_{\pm.02}$	$\mathbf{.94}_{\pm.02}$	$\underline{.79}_{\pm.05}$	$\mathbf{.82}_{\pm.03}$	$.73_{\pm.03}$	$\underline{.79}_{\pm.02}$
3	4	$.87_{\pm.02}$	$\underline{.63}_{\pm.07}$	$.33_{\pm.08}$	$.59_{\pm.05}$	$\mathbf{.69}_{\pm.06}$	$\underline{.60}_{\pm.05}$	$.31_{\pm.07}$	$.54_{\pm.04}$	$\mathbf{.62}_{\pm.05}$
	10	$.86_{\pm.05}$	$.81_{\pm.07}$	$.58_{\pm.13}$	$\mathbf{.88}_{\pm.04}$	$\underline{.87}_{\pm.05}$	$\mathbf{.76}_{\pm.04}$	$.53_{\pm.12}$	$\mathbf{.76}_{\pm.02}$	$\underline{.75}_{\pm.04}$
	40	$.91_{\pm.01}$	$\underline{.95}_{\pm.01}$	$.91_{\pm.01}$	$\mathbf{.97}_{\pm.00}$	$\underline{.95}_{\pm.03}$	$.74_{\pm.03}$	$\mathbf{.84}_{\pm.03}$	$\underline{.80}_{\pm.01}$	$\underline{.80}_{\pm.03}$
4	4	$.80_{\pm.02}$	$\mathbf{.87}_{\pm.04}$	$\underline{.77}_{\pm.06}$	$.68_{\pm.05}$	$\underline{.77}_{\pm.05}$	$\mathbf{.77}_{\pm.04}$	$\underline{.72}_{\pm.05}$	$.64_{\pm.05}$	$\underline{.72}_{\pm.03}$
	10	$.80_{\pm.04}$	$\mathbf{.95}_{\pm.01}$	$\underline{.94}_{\pm.01}$	$.87_{\pm.03}$	$.91_{\pm.02}$	$\underline{.83}_{\pm.01}$	$\mathbf{.84}_{\pm.01}$	$.77_{\pm.02}$	$.81_{\pm.02}$
	40	$.84_{\pm.02}$	$\mathbf{.96}_{\pm.01}$	$\underline{.94}_{\pm.01}$	$.92_{\pm.03}$	$\underline{.94}_{\pm.01}$	$\mathbf{.85}_{\pm.02}$	$\underline{.84}_{\pm.01}$	$.78_{\pm.02}$	$.81_{\pm.05}$
5	4	$.02_{\pm.00}$	$\underline{.56}_{\pm.07}$	$.41_{\pm.07}$	$.52_{\pm.06}$	$\mathbf{.65}_{\pm.06}$	$\mathbf{.54}_{\pm.06}$	$.42_{\pm.06}$	$.40_{\pm.04}$	$\underline{.48}_{\pm.04}$
	10	$.01_{\pm.00}$	$\underline{.79}_{\pm.05}$	$.78_{\pm.04}$	$.78_{\pm.06}$	$\mathbf{.90}_{\pm.03}$	$\mathbf{.73}_{\pm.04}$	$\underline{.69}_{\pm.06}$	$.58_{\pm.05}$	$.65_{\pm.03}$
	40	$.01_{\pm.01}$	$.82_{\pm.06}$	$\mathbf{.94}_{\pm.00}$	$\underline{.90}_{\pm.05}$	$\underline{.90}_{\pm.05}$	$\underline{.76}_{\pm.02}$	$\mathbf{.85}_{\pm.04}$	$.60_{\pm.06}$	$.73_{\pm.10}$

B.5 COMPUTE

Landings and grafts ran on one NVIDIA A40 (48 GB) per job for the Qwen prompt lock and the Llama and Mistral locks, and on one NVIDIA L40S (48 GB) per job for the five Qwen LoRA locks. Fine-tuning ran on L40S GPUs, the honest activations were captured on one L40S, and the honest graft ran on A40 GPUs. The time to fit the shift in Section 5.1 is the wall-clock time of the landing stage, summed over layers and batches, on the four Llama and Mistral locks at $k \in \{4, 10, 40\}$, without model loading or grafting. For each locked model and k, the three ActFlow rules ran together in one job and GD in another job on the same machine. GD took 22.8 GPU-hours and the joint job of the three ActFlow rules 11.1. Counting one rule as a third of that job, GD took 6.2 times as long as one ActFlow rule. The ratio for one (locked model, k) pair ranges from 2.9, where GD stops early, to 8.7, where GD runs all 1000 steps. One ActFlow step evaluates $F_{S,\ell,K}$ and $J_{S,\ell,K}$ with one forward pass and q backward passes through the layers after ℓ, and one GD step evaluates $F_{S,\ell,K}$ and $J_{S,\ell,K}^{\top}(\gamma_{H,K} - F_{S,\ell,K})$ with one forward pass and one backward pass. With $N = 40$, one ActFlow rule makes 41 Jacobian evaluations, one of them at the final point for the per-step record, and GD makes up to 1001 gradient evaluations, the last of them the stopping check after step 1000. Table 6 gives the GPU-hours of every stage, 130 in total.

Table 6: GPU-hours of each stage of the experiments, as the wall-clock time of each job summed over jobs, with one job per GPU. The first stage includes checks that the code reproduces the locked logits. ActFlow runs include every rule and config of Appendix B.2, and grafts include layer selection and the evaluation on both test sets. Model downloads and idle time are not counted.

stage	A40	L40S	total
Locked activations and accuracies	0.25	0.17	0.43
ActFlow runs	22.50	11.35	33.85
GD runs	27.84	8.32	36.17
Grafts of the ActFlow shifts	27.47	12.59	40.06
Grafts of the GD shifts	4.20	1.90	6.09
Honest activations	–	0.10	0.10
Honest graft	5.21	–	5.21
Fine-tuning	–	8.46	8.46
Total	87.47	42.89	130.36

B.6 FULL RESULTS

Table 7 gives the test accuracy of ActFlow_5 and $\text{ActFlow}_{\text{PR}}$ for the configs of Figure 3c at every k. Table 8 gives the accuracy of the graft on the k labeled items and the landing error of each method at the chosen layer, for all ten locked models. Table 9 gives the chosen layers, and Table 10 the balanced accuracy and the share of the most frequent answer letter on the ARC-Easy test items. Table 11 gives the results of $\text{ActFlow}_{0.1}$, which we ran only on the two Qwen locks of Table 2. Table 12 gives the landing error of $\text{ActFlow}_{\text{full}}$ at the last layer. At every $k \geq 4$ the rank of $J_{S,n_{\text{L}},K}(0)$ is $q+k$ in every draw, and no method brings any item within 0.5 of its target at the last layer, except GD for one item on the Qwen prompt lock at $k = 4$. Table 13 compares the chosen layers with the window of Tan et al. (2026). On the eight locked models with a window, $\hat{\ell}$ lies in it in 103 (ActFlow_5), 88 ($\text{ActFlow}_{\text{PR}}$), 112 ($\text{ActFlow}_{\text{full}}$) and 114 (GD) of 240 draws, and in 224 for the honest graft.

Table 7: ARC-Easy test accuracy of ActFlow_5 and $\text{ActFlow}_{\text{PR}}$ on the two Qwen locks for the configs of Figure 3c at $k = 4$, 10 and 40. GN is the Gauss–Newton correction and seq. the sequential path. Subscripts are standard errors over draws.

lock	rule	k	GN N=1	GN N=8	GN N=40	Euler N=40	seq. N=40
prompt	AF_5	4	$.59_{\pm.05}$	$.68_{\pm.05}$	$.73_{\pm.06}$	$.57_{\pm.06}$	$.77_{\pm.05}$
		10	$.66_{\pm.06}$	$.84_{\pm.06}$	$.92_{\pm.02}$	$.72_{\pm.08}$	$.89_{\pm.04}$
		40	$.80_{\pm.01}$	$.93_{\pm.01}$	$.95_{\pm.00}$	$.78_{\pm.03}$	$.92_{\pm.00}$
	AF_{PR}	4	$.51_{\pm.05}$	$.66_{\pm.06}$	$.74_{\pm.06}$	$.49_{\pm.06}$	$.79_{\pm.04}$
		10	$.68_{\pm.06}$	$.80_{\pm.07}$	$.88_{\pm.03}$	$.69_{\pm.08}$	$.85_{\pm.05}$
		40	$.69_{\pm.03}$	$.93_{\pm.03}$	$.95_{\pm.01}$	$.83_{\pm.02}$	$.92_{\pm.01}$
LoRA	AF_5	4	$.02_{\pm.01}$	$.18_{\pm.06}$	$.32_{\pm.08}$	$.02_{\pm.01}$	$.50_{\pm.08}$
		10	$.01_{\pm.00}$	$.14_{\pm.12}$	$.61_{\pm.15}$	$.01_{\pm.00}$	$.74_{\pm.12}$
		40	$.01_{\pm.01}$	$.01_{\pm.01}$	$.93_{\pm.03}$	$.01_{\pm.00}$	$.76_{\pm.21}$
	AF_{PR}	4	$.00_{\pm.00}$	$.01_{\pm.00}$	$.03_{\pm.01}$	$.01_{\pm.00}$	$.03_{\pm.01}$
		10	$.00_{\pm.00}$	$.01_{\pm.01}$	$.10_{\pm.08}$	$.00_{\pm.00}$	$.35_{\pm.14}$
		40	$.00_{\pm.00}$	$.01_{\pm.01}$	$.37_{\pm.34}$	$.00_{\pm.00}$	$.24_{\pm.18}$

Table 8: Fit on the k labeled items. Accuracy is the share of the k labeled items that the graft at the chosen layer $\hat{\ell}$ answers correctly, averaged over draws. Error is the median over the 80 pool items of $\|F_{S,\hat{\ell},i}(h_{S,\hat{\ell},i} + x_N) - \gamma_{H,i}\|_\infty$, the largest distance of the four logits of item i from its target after the last step at the layer $\hat{\ell}$ of its draw, before the graft. Dashes mark ActFlow_5 at $k = 1$, where it equals $\text{ActFlow}_{\text{full}}$, and the honest graft at $k = 1$, which was not run.

				accuracy on the labeled items					error at $\hat{\ell}$			
model	lock	seed	k	AF_5	AF_{PR}	AF_{full}	GD	honest graft	AF_5	AF_{PR}	AF_{full}	GD
Qwen2.5-7B	prompt		1	–	.79	.89	.99	–	–	3.69	0.01	0.49
			4	.86	.84	.99	1.00	.95	5.19	5.62	0.01	0.35
			10	.89	.89	.99	1.00	.93	5.80	6.19	0.01	0.27
			40	.91	.91	.98	1.00	.91	7.02	6.91	0.01	1.00
	LoRA	1	1	–	.10	.45	.38	–	–	8.23	0.01	0.49
			4	.46	.05	.99	.91	.99	5.58	9.04	0.01	0.25
			10	.66	.15	1.00	1.00	.98	5.13	8.29	0.01	0.19
			40	.93	.40	1.00	1.00	.95	5.09	6.16	0.01	0.56
	LoRA	2	4	.93	.70	.91	.94	.99	5.42	6.58	0.01	0.33
			10	.94	.91	1.00	1.00	.99	5.90	5.83	0.01	0.24
			40	.94	.93	1.00	1.00	.98	6.08	6.15	0.01	0.64
	LoRA	3	4	.88	.43	.99	1.00	.94	3.71	6.23	0.01	0.34
			10	.94	.60	1.00	1.00	.93	4.31	5.09	0.01	0.29
			40	.94	.94	1.00	1.00	.91	3.52	3.90	0.02	0.62
	LoRA	4	4	.95	.90	.96	.99	.93	4.94	5.66	0.01	0.38
			10	1.00	.99	1.00	1.00	.89	5.60	5.65	0.01	0.20
			40	.98	.98	1.00	1.00	.83	4.64	4.36	0.01	0.45
	LoRA	5	4	.79	.56	.98	.98	.08	3.81	5.67	0.01	0.30
			10	.86	.86	1.00	1.00	.09	5.35	4.32	0.01	0.35
			40	.76	.91	1.00	1.00	.08	6.43	4.73	0.01	1.10
Llama-3-8B	prompt		4	.95	.80	1.00	1.00	.96	1.75	2.78	0.00	0.34
			10	.94	.93	1.00	1.00	.96	2.30	2.51	0.00	0.42
			40	.95	.94	1.00	.99	.95	2.42	2.71	0.00	1.15
	LoRA	1	4	.93	.84	.86	.98	.94	4.66	6.77	0.02	0.32
			10	.93	.88	.91	.99	.95	6.17	7.96	0.01	0.28
			40	.90	.95	.96	.99	.94	6.94	7.70	0.01	0.09
Mistral-7B	prompt		4	.88	.69	.90	1.00	.79	5.67	7.27	0.02	0.37
			10	.75	.74	.85	.99	.80	7.31	7.87	0.01	0.37
			40	.69	.71	.93	.98	.78	9.35	8.19	0.02	0.49
	LoRA	1	4	.61	.49	.84	.88	.11	5.76	7.78	0.01	0.27
			10	.66	.74	.88	.89	.10	6.13	5.03	0.01	0.28
			40	.66	.83	.65	.83	.09	6.66	5.31	0.00	0.15

Table 9: Chosen layer $\hat{\ell}$ as the median over draws, with the smallest and largest $\hat{\ell}$ in parentheses when they differ. Layers are numbered from 1 to n_{L}, which is 28 for Qwen and 32 for Llama and Mistral. Dashes are as in Table 8.

model	lock	seed	k	AF_5	$\mathrm{AF}_{\mathrm{PR}}$	$\mathrm{AF}_{\mathrm{full}}$	GD	honest graft
Qwen2.5-7B	prompt		1	–	9 (1–19)	13.5 (1–22)	9 (1–22)	–
			4	16 (7–17)	12.5 (2–19)	16 (10–18)	13.5 (9–17)	19 (16–20)
			10	17 (12–18)	16 (7–18)	18 (14–19)	13.5 (10–19)	19 (16–19)
			40	17 (16–18)	17.5 (17–18)	24	19.5 (19–20)	19
	LoRA	1	1	–	10 (1–25)	19 (1–24)	18 (2–23)	–
			4	15.5 (1–27)	11 (1–27)	19 (16–24)	18 (15–25)	16 (15–20)
			10	16 (13–26)	10 (1–21)	19 (18–24)	17 (15–25)	16 (16–22)
			40	17	11 (10–12)	22 (21–23)	20.5 (18–23)	16
	LoRA	2	4	14 (3–20)	6 (2–18)	23 (10–25)	14.5 (4–24)	15.5 (8–16)
			10	10 (4–18)	14.5 (6–20)	20.5 (14–25)	19 (14–24)	16 (8–16)
			40	14 (10–18)	15 (12–18)	24	23.5 (23–24)	12 (8–16)
	LoRA	3	4	14 (1–22)	19.5 (2–28)	18.5 (15–24)	18 (10–25)	11 (10–12)
			10	17 (3–20)	19 (12–24)	21 (17–23)	19.5 (15–23)	11 (11–17)
			40	16 (8–24)	17.5 (17–18)	21 (20–22)	20.5 (17–24)	11
	LoRA	4	4	16 (3–18)	4 (2–17)	17 (16–26)	14.5 (2–24)	12 (12–19)
			10	17.5 (2–22)	13 (3–18)	16.5 (16–17)	17 (17–18)	12 (12–19)
			40	16.5 (16–17)	17 (16–18)	19.5 (18–21)	17.5 (17–18)	14.5 (12–17)
	LoRA	5	4	15.5 (5–22)	12 (2–20)	22.5 (15–26)	23 (14–24)	22 (10–28)
			10	14 (5–21)	15.5 (5–18)	20.5 (17–25)	24 (23–25)	10 (8–28)
			40	10 (9–11)	10	25.5 (25–26)	22.5 (21–24)	9 (8–10)
Llama-3-8B	prompt		4	13 (3–32)	13 (9–15)	8.5 (7–30)	10.5 (4–16)	15 (13–18)
			10	13 (7–13)	13 (12–13)	8.5 (7–12)	13 (8–16)	13 (11–17)
			40	13	12.5 (12–13)	13.5 (11–16)	14.5 (13–16)	13
	LoRA	1	4	5.5 (2–11)	6 (2–15)	6 (2–31)	4.5 (1–14)	13.5 (12–18)
			10	5.5 (3–10)	5 (5–8)	13 (9–19)	11 (1–19)	13.5 (11–16)
			40	5.5 (4–7)	12.5 (6–19)	20 (10–30)	15.5 (11–20)	16
Mistral-7B	prompt		4	9.5 (1–32)	7 (1–32)	11 (5–31)	11 (5–14)	13 (11–16)
			10	8 (5–13)	7.5 (5–13)	25 (12–31)	13 (11–18)	13 (12–13)
			40	11.5 (11–12)	13.5 (11–16)	30	17 (12–22)	12.5 (12–13)
	LoRA	1	4	6 (1–16)	12.5 (1–30)	11.5 (1–31)	11.5 (6–31)	32 (9–32)
			10	13 (1–18)	12.5 (1–16)	14 (12–31)	13 (10–31)	32 (18–32)
			40	16.5 (16–17)	15.5 (15–16)	29 (28–30)	20.5 (16–25)	26.5 (21–32)

Table 10: Balanced accuracy and top share on the 100 ARC-Easy test items, averaged over draws. Balanced accuracy is the mean over the four correct letters of the accuracy on the test items with that correct letter. Top share is the share of test answers that are the most frequent answer letter of the draw. The correct letters A, B, C and D have shares .26, .32, .22 and .20 among the test items. Dashes are as in Table 8.

				balanced accuracy				top share			
model	lock	seed	k	AF_5	$\mathrm{AF}_{\mathrm{PR}}$	$\mathrm{AF}_{\mathrm{full}}$	GD	AF_5	$\mathrm{AF}_{\mathrm{PR}}$	$\mathrm{AF}_{\mathrm{full}}$	GD
Qwen2.5-7B	prompt		1	–	.43	.40	.43	–	.45	.48	.45
			4	.73	.73	.51	.81	.39	.40	.47	.36
			10	.92	.88	.58	.87	.32	.35	.39	.34
			40	.95	.95	.63	.82	.32	.32	.34	.33
	LoRA	1	1	–	.04	.16	.17	–	.36	.66	.61
			4	.33	.02	.55	.62	.50	.34	.45	.41
			10	.62	.11	.79	.85	.35	.37	.35	.34
			40	.94	.36	.92	.94	.30	.30	.33	.29
	LoRA	2	4	.75	.60	.46	.66	.39	.40	.45	.43
			10	.78	.75	.72	.76	.38	.39	.40	.41
			40	.87	.81	.92	.94	.33	.35	.31	.31
	LoRA	3	4	.65	.34	.60	.71	.45	.37	.40	.38
			10	.82	.60	.89	.89	.32	.39	.35	.34
			40	.96	.91	.97	.95	.31	.33	.31	.31
	LoRA	4	4	.88	.78	.67	.78	.30	.30	.40	.35
			10	.95	.95	.87	.91	.30	.29	.32	.31
			40	.96	.95	.92	.94	.31	.28	.30	.31
	LoRA	5	4	.59	.44	.53	.66	.35	.34	.46	.40
			10	.81	.80	.78	.89	.32	.33	.38	.36
			40	.83	.95	.90	.89	.30	.28	.35	.34
Llama-3-8B	prompt		4	.83	.79	.74	.81	.40	.40	.43	.39
			10	.91	.90	.82	.82	.34	.34	.36	.38
			40	.92	.91	.70	.88	.34	.33	.39	.35
	LoRA	1	4	.79	.80	.53	.77	.31	.32	.35	.33
			10	.81	.86	.50	.71	.35	.31	.39	.33
			40	.89	.91	.70	.72	.33	.30	.41	.35
Mistral-7B	prompt		4	.54	.42	.40	.63	.54	.55	.52	.44
			10	.61	.54	.22	.63	.45	.45	.48	.43
			40	.69	.74	.32	.54	.45	.43	.43	.40
	LoRA	1	4	.31	.24	.30	.32	.56	.48	.47	.57
			10	.47	.54	.40	.45	.41	.44	.38	.36
			40	.74	.79	.40	.52	.37	.32	.46	.41

Table 11: $\mathrm{ActFlow}_{0.1}$ on the two Qwen2.5-7B locks of Table 2. Test accuracies are on 100 ARC-Easy and 100 OpenBookQA items with standard errors over draws as subscripts, and the other columns are as in Tables 8, 9 and 10.

lock	k	ARC-Easy	OpenBookQA	labeled accuracy	error at $\hat{\ell}$	$\hat{\ell}$	balanced accuracy	top share
prompt	1	$.41_{\pm.03}$	$.31_{\pm.02}$	.90	0.01	13 (1–22)	.39	.48
	4	$.65_{\pm.05}$	$.46_{\pm.03}$	.98	1.32	15 (1–19)	.64	.40
	10	$.76_{\pm.04}$	$.55_{\pm.05}$	.96	3.07	15 (12–19)	.76	.36
	40	$.84_{\pm.09}$	$.56_{\pm.16}$	.86	3.65	18 (17–19)	.84	.33
LoRA	1	$.17_{\pm.01}$	$.21_{\pm.01}$	.44	0.01	19 (1–24)	.16	.67
	4	$.55_{\pm.06}$	$.54_{\pm.05}$	.99	0.02	19 (14–24)	.56	.45
	10	$.83_{\pm.04}$	$.76_{\pm.03}$	1.00	0.86	18.5 (15–22)	.85	.36
	40	$.91_{\pm.01}$	$.83_{\pm.00}$	.98	2.26	17	.92	.26

Table 12: The last layer $\ell = n_{\mathrm{L}}$. Entries are the largest landing error $\max_{i \in K} \|F_{S,n_{\mathrm{L}},i}(h_{S,n_{\mathrm{L}},i} + x) - \gamma_{H,i}\|_\infty$ of the k items after $\mathrm{ActFlow}_{\mathrm{full}}$ at the last layer, as medians over draws, with the smallest value over draws in parentheses. After $\mathrm{ActFlow}_{\mathrm{full}}$ at the last layer, no item of any draw has all q logits within 0.5 of its target.

model	lock	seed	$k = 4$	$k = 10$	$k = 40$
Qwen2.5-7B	prompt		10.3 (5.3)	13.9 (11.8)	16.7 (16.5)
	LoRA	1	10.3 (7.0)	12.1 (11.3)	12.5 (11.9)
	LoRA	2	14.2 (7.5)	16.4 (14.4)	25.9 (13.7)
	LoRA	3	12.5 (7.1)	15.6 (13.1)	15.0 (14.7)
	LoRA	4	12.2 (7.2)	13.3 (12.2)	13.6 (13.3)
	LoRA	5	12.5 (8.1)	13.7 (11.6)	17.0 (12.6)
Llama-3-8B	prompt		5.1 (2.9)	5.7 (5.3)	6.5 (6.0)
	LoRA	1	14.0 (7.8)	19.8 (16.3)	23.0 (20.5)
Mistral-7B	prompt		13.9 (5.8)	16.7 (15.4)	18.5 (17.9)
	LoRA	1	11.4 (8.2)	11.8 (9.6)	11.0 (10.5)

Table 13: Chosen layers $\hat{\ell}$ inside the window of Tan et al. (2026), the layers at which their single-layer graft recovers at least 0.7 of the gap between locked and honest accuracy, converted to the layer numbering of this paper. Their graft was run only at layers 10 to 28 of Qwen and 11 to 32 of Llama and Mistral, so a window that starts at layer 10 or 11 may extend to lower layers, and none marks a lock on which no tested layer reaches 0.7. Entries count the draws at $k = 4$, 10 and 40 (30 per method) whose $\hat{\ell}$ lies in the window.

model	lock	seed	window	AF_5	$\mathrm{AF}_{\mathrm{PR}}$	$\mathrm{AF}_{\mathrm{full}}$	GD	honest graft
Qwen2.5-7B	prompt		15–20	19	14	22	13	30
	LoRA	1	10–22	21	16	21	21	30
	LoRA	2	10–26	21	13	30	28	23
	LoRA	3	11–12	1	2	0	2	28
	LoRA	4	12, 17–18	11	6	14	13	27
	LoRA	5	none	–	–	–	–	–
Llama-3-8B	prompt		11–17	21	28	10	19	29
	LoRA	1	12–20	0	3	8	5	29
Mistral-7B	prompt		12–15	9	6	7	13	28
	LoRA	1	none	–	–	–	–	–