Secret Coordinates: Keyed Symmetry Gauges for Edit-Resistant Open Weights

Ching Lam Choi (chinglam@mit.edu)

Abstract

Open-weight models reveal not only a function, but a coordinate system in which downstream adaptation methods operate. This project investigates whether exact parameter symmetries can be used to define keyed reparameterisations that preserve the released model's function while disrupting unauthorised coordinate-local editing. The objective is not confidentiality of the model function, but a measurable separation between the cost of authorised editing with the key and unauthorised editing in the exposed coordinates. We formalise this objective through an edit-forgery game, prove a separation for diagonal hidden-feature edits in a two-layer linear model, analyse a block- and head-local extension motivated by transformer adapters, and study non-orthogonal symmetries as a source of optimiser-bounded difficulty. We find that keyed symmetries can obstruct edit families whose efficiency depends on coordinate locality, but they do not generically increase the rank required by unrestricted dense updates.

1 Introduction

Open-weight models are valuable because they are auditable but also *editable*: once parameters are released, users can fine-tune for new domains, attach adapters, merge checkpoints, apply task vectors, or perform targeted edits. Editability enables research, customisation, auditing, and deployment, but also changes the security considerations of a model release. A third party can cheaply adapt a model, alter its safety behaviour, specialise it for harmful tasks, or distribute small adapters that substantially change behaviour while leaving the original checkpoint mostly intact. Fine-tuning has been shown to compromise aligned language models [Qi et al., 2024], motivating work on tamper-resistant safeguards and safer adaptation [Tamirisa et al., 2025, Hsu et al., 2024]. Many of the relevant methods—LoRA and QLoRA [Hu et al., 2022, Dettmers et al., 2023], IA^3 [Liu et al., 2022], DoRA [Liu et al., 2024], task arithmetic [Ilharco et al., 2023], model soups and merging [Wortsman et al., 2022, Goddard et al., 2024], and ROME-style editing [Meng et al., 2022]—derive their efficiency from structure in the released coordinates: edits are often *diagonal, sparse, low-rank, head-local, neuron-local, or module-local*. An open-weight release therefore exposes not only a function, but also a coordinate system in which cheap edits are defined.

This project begins with the fact that releasing weights means choosing one parameterisation among many functionally equivalent ones. Neural networks often have parameter symmetries: transformations of the weights that preserve the input-output function. Hidden units can be permuted; intermediate representations can sometimes be re-based with compensating transformations in adjacent layers; attention projections admit certain basis changes; and normalisation or scaling symmetries can alter the geometry of parameter space while preserving the computed function [Hecht-Nielsen, 1990, Godfrey et al., 2022, Lim et al., 2024, Lim, 2025]. A model function therefore corresponds not to a single point in parameter space, but to an equivalence class of parameter settings. We study

whether a releaser can choose a representative of that class with a secret key: the published model behaves identically at inference time, but subsequent weight-space edits must be expressed in a transformed coordinate system. Authorised users who know the key can invert the transformation, edit in canonical coordinates, and re-apply it; unauthorised users see only the transformed weights. The goal of secret coordinates is to hide cheap edit structure, not the model function.

The security objective is therefore *edit-cost separation*. An authorised user should be able to perform a structured edit at low cost, while an unauthorised user should need a larger rank, a denser adapter, more data, more optimisation steps, better preconditioning, or an explicit canonicalisation attack to achieve the same effect. Keyed symmetries can obstruct edit families whose efficiency depends on coordinate locality, while dense or equivariant edit families define the natural boundary of the approach. In short, secret coordinates break locality, not rank.

1.1 Contributions

This report makes the following contributions: (i) we formulate secret-coordinate releases as a keyed edit-access-control mechanism and define an edit-forgery security game parameterised by the adversary's edit family and budget; (ii) we prove a clean separation for diagonal hidden-feature edits in a two-layer linear model, showing that a random hidden basis makes exposed-basis diagonal editing capture only a vanishing fraction of the non-scalar edit energy; (iii) we extend the perspective to block- and head-local edits, giving the transformer-motivated analogue in which the key buys locality and unkeyed editors must move to cross-block updates; (iv) we analyse non-orthogonal symmetries as an optimiser-bounded obstruction rather than a representational one; and (v) we identify the main boundary attacks—dense LoRA, full fine-tuning, and canonicalisation—that delimit when symmetry-based edit resistance can and cannot apply.

2 Background

2.1 Parameter Symmetries

Let $F : \Theta \to \mathcal{F}$ map parameters to the functions they compute. A group action $G \curvearrowright \Theta$ is a parameter symmetry if $F(g \cdot \theta) = F(\theta)$ for every $g \in G$ and $\theta \in \Theta$; they are common in neural networks. In a multilayer perceptron, hidden units can be permuted if the incoming and outgoing weights are permuted consistently. In a linear network, an intermediate representation can be expressed in a different basis by multiplying one layer by an invertible matrix and the next by its inverse. In attention, certain changes of basis in the query–key and value–output pathways preserve dot products or cancel across adjacent projections. Other examples include head permutations, scaling symmetries induced by homogeneous nonlinearities or normalisation layers, and basis changes in state-space models [Hecht-Nielsen, 1990, Godfrey et al., 2022, Lim et al., 2024, Lim, 2025].

A model's input-output behaviour does not uniquely determine its coordinates in parameter space. Two parameter settings can compute exactly the same function while having very different weights, distances to other checkpoints, gradient geometry, or apparent local structure [Ainsworth et al., 2023, Nagarajan and Kolter, 2019, Frankle et al., 2020]. This distinction is usually treated as a nuisance for optimisation, model comparison, or merging; here, it composes the foundation for our work. A model releaser does not merely decide whether to release a function. Conditional on releasing open weights, the releaser also chooses a representative of a symmetry class. This choice can affect downstream weight-space operations even when it has no effect on inference.

2.2 Coordinate-local editing

Many open-weight editing methods are efficient because they exploit structure in the released coordinates. IA^3 and related gating methods apply coordinate-wise or diagonal rescalings. LoRA, QLoRA, and DoRA restrict updates to low-rank or weight-decomposed forms in particular matrices. Head-local or module-local adapters assume that specific architectural components remain meaningful editing units. ROME-style interventions target selected modules and activation directions. Task arithmetic and model merging rely on the assumption that corresponding parameters across checkpoints are aligned well enough for addition or averaging to be meaningful. These methods are not merely using the function represented by the model; they are using the coordinate system exposed by the release.

We distinguish between edit families that are coordinate-local and edit families that are basis-equivariant (or, dense). A coordinate-local method is cheap because it restricts attention to a special structure in the exposed basis: diagonal entries, sparse neurons, individual heads, blocks, modules, or low-dimensional subspaces tied to named weights. A basis-equivariant or dense method is less tied to that basis: for example, a full dense update or sufficiently expressive dense low-rank update may remain representable after an invertible change of coordinates. Secret coordinates are therefore not expected to obstruct all possible edits. They are designed to target edit families whose efficiency depends on locality or simplicity in the released parameterisation.

2.3 Cryptographic interpretation

The cryptographic intuition underlying this project is not semantic encryption. The released model is intended to remain usable, and its input-output function is public by design. Instead, the key selects a gauge: a function-preserving coordinate representation of the model. An authorised user who knows the key can invert the gauge, perform an edit in canonical coordinates, and re-apply the gauge. An unauthorised user receives the same functional model but must express edits in the released coordinates. If the useful edit is simple only in the hidden coordinate system, then a standard edit method in the exposed coordinates may require a larger or denser update, more optimisation effort, or an explicit attempt to recover the gauge.

This shifts the security target from confidentiality to edit access control. The adversary is not trying to decrypt the model function; the adversary is trying to produce a cheap edited model. Cryptanalysis therefore means recovering, approximating, or bypassing the hidden gauge, for example by aligning the released weights to a public reference checkpoint, using invariant or equivariant edit methods, or abandoning coordinate-local edits in favour of denser updates [Ainsworth et al., 2023, Lim et al., 2024]. The appropriate security question is consequently an edit-forgery question: for a specified edit family and budget, how much more costly is it to achieve the target edit without the key than with the key? This framing is deliberately narrower than general model security, but it gives a precise language for reasoning about when parameter symmetries can provide meaningful resistance to cheap unauthorised weight-space modification.

3 Secret Coordinate Releases and the Edit-Forgery Game

3.1 Release mechanism

A secret-coordinate release samples a key $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, derives a symmetry $g_k \in G$, and publishes $\theta' = g_k \cdot \theta$. Since G acts by parameter symmetries, the release satisfies $F(\theta') = F(\theta)$: the weights have been reparameterised, but the input-output function is unchanged. This is the

Algorithm 1 $\mathsf{EditForge}_{\mathcal{M},G,\mathcal{T},\mathcal{U},B,\tau}$

1: Challenger samples $\theta \in \mathcal{M}$, $k \leftarrow \mathsf{KeyGen}(1^\lambda)$, and $g_k \in G$.
2: Challenger releases $\theta' = g_k \cdot \theta$ and edit task $\mathcal{D} \sim \mathcal{T}$.
3: Authorised editor, knowing k, chooses $a \in \mathcal{A}_k$ with cost $C(a)$.
4: Adversary, not knowing k, chooses $u \in \mathcal{U}$ with cost $C(u) \le B$.
5: Adversary wins if $\mathrm{Score}(u(\theta'), \mathcal{D}) \ge \tau$.
6: Define

$$A_{\mathcal{U}}(\tau) = \frac{C^{\mathcal{U}}_{\mathrm{unauth}}(\tau)}{C_{\mathrm{auth}}(\tau)}.$$

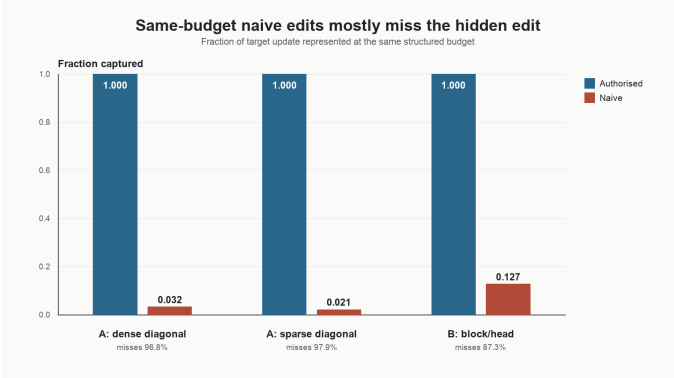


Figure 1: Authorised editors use the keyed local family and capture the target exactly; same-budget naive adversaries edit in the released basis and capture only a small fraction of the canonical edit.

basic distinction from encryption. The released model remains usable and inspectable; only the coordinate system in which edits are simple has been changed.

An authorised editor who knows k can work in the canonical coordinates, apply an edit there, and push the edited model back into the released coordinates. Schematically,

$$\theta'_{\mathrm{edit}} = g_k \cdot \big(g_k^{-1} \cdot \theta' + \Delta\big). \tag{1}$$

The pushforward of Δ depends on the model class and edit family. In linear settings below, hidden-space edits are conjugated by the secret basis change. The authorised editor knows the basis in which the intended edit is cheap, while an unauthorised editor sees only the exposed coordinates.

3.2 Edit-Forgery Adversaries and Scope

Algorithm 1 formalises the security objective. The adversary is not trying to recover the model function, which is public by design; it is trying to produce a cheap edited model from the released weights. The relevant security quantity is the edit-cost separation

$$A_{\mathcal{U}}(\tau) = \frac{C^{\mathcal{U}}_{\mathrm{unauth}}(\tau)}{C_{\mathrm{auth}}(\tau)}, \tag{2}$$

where $C_{\mathrm{auth}}(\tau)$ is the least cost for a keyed editor to reach task threshold τ, and $C^{\mathcal{U}}_{\mathrm{unauth}}(\tau)$ is the least cost for an unkeyed editor restricted to edit family $\mathcal{U}$. A release is edit-resistant against $\mathcal{U}$ when this ratio is large. The dependence on $\mathcal{U}$ is essential: the same release may obstruct exposed-basis diagonal gates, partially obstruct head-local adapters, and provide no algebraic barrier to sufficiently expressive dense updates.

We consider three adversary levels. The *standard-tool adversary* (L0) does not know that a symmetry was applied and runs existing PEFT, editing or merging tools directly on θ'. The *aware but unkeyed adversary* (L1) knows the release mechanism and G, but not k; it may increase rank, use larger adapters, train with more steps or samples, or move from local to cross-coordinate updates. The *cryptanalytic adversary* (L2) attempts to recover or bypass the hidden gauge, *e.g.*, by solving

$$\widehat{g} = \arg\min_{g \in G} d(g^{-1} \cdot \theta', \theta_{\mathrm{ref}}) \tag{3}$$

for a public reference model or related checkpoint, and then editing in the recovered gauge.

The defended classes are *coordinate-local or optimiser-bounded edit families:* diagonal or IA^3-style gates, sparse neuron edits, block- or head-local adapters, module-local updates, and practical PEFT recipes constrained by rank, data, precision, or optimisation budget. Unaddressed "boundary" attacks include edits that are invariant or equivariant to the hidden basis, or expressive enough to ignore locality. Examples include dense LoRA at sufficient rank, full fine-tuning, prompt attacks, activation steering, distillation, and key leakage. We proffer that secret-coordinate releases create cost separation against edit methods whose efficiency depends on the exposed coordinate system. It does not prevent all possible modifications of an open-weight model.

4 Hidden-Coordinate Locality Assumptions

This section studies the locality assumptions that secret coordinates disrupt. Settings A and B concern representational locality: the intended edit is simple in the keyed basis but delocalised in the released basis. Setting C concerns optimiser locality: the edit may remain representable, but the released coordinates can make standard training recipes poorly conditioned. Together, these settings separate three effects: diagonal locality, block/head locality, and optimiser friendliness.

4.1 Hidden Basis vs. Diagonal Edits

Consider a two-layer linear model $f_\theta(x) = BAx$, where $A \in \mathbb{R}^{d\times n}$ maps inputs into a d-dimensional hidden space and $B \in \mathbb{R}^{m\times d}$ maps hidden states to outputs. For any orthogonal $R \in O(d)$, release $A' = RA$ and $B' = BR^\top$. Then $B'A' = BR^\top RA = BA$, so the released model computes exactly the same function. The secret key is the hidden-basis choice R.

Now suppose the useful authorised edit is a diagonal hidden-feature gate, $f_D(x) = B(I + D)Ax$, where $D = \mathrm{diag}(d_1, \ldots, d_d)$. This is a toy model for IA^3-style multiplicative edits, feature gates, sparse semantic-feature edits, or neuron-local interventions. In released coordinates, the same edited function is implemented as $B'(I + RDR^\top)A'$. Thus the authorised keyed edit family is $\mathcal{A}_k = \{RDR^\top : D \text{ diagonal}\}$: it has only d degrees of freedom, but appears dense in the released basis. By contrast, a naive exposed-basis diagonal adversary uses $\mathcal{U}_{\mathrm{diag}} = \{H : H \text{ diagonal in the released basis}\}$. So the adversary is trying to approximate $RDR^\top$ by a diagonal matrix in the wrong basis.

For an exposed-coordinate target update $M_\star$, define the capture and residual of an adversarial approximation $\widehat{M}$ by

$$\text{Capture} = 1 - \frac{\|M_\star - \widehat{M}\|_F^2}{\|M_\star\|_F^2}, \qquad \text{Residual} = \frac{\|M_\star - \widehat{M}\|_F^2}{\|M_\star\|_F^2}. \tag{4}$$

Under whitened hidden activations and $B^\top B = I$, *this Frobenius projection error is exactly the population squared loss of the edited model.* More generally, it becomes a weighted Frobenius norm; we give the full derivation in Appendix B. We say that *a random hidden basis makes a trace-free diagonal edit almost entirely non-diagonal in the exposed basis.*

Theorem 4.1 (Hidden-basis diagonal separation)**.** *Let $D_0 \in \mathbb{R}^{d\times d}$ be diagonal and trace-free, and let $R \sim \mathrm{Haar}(O(d))$. Then*

$$\mathbb{E}_R\Big[\|\Pi_{\mathrm{diag}}(RD_0R^\top)\|_F^2\Big] = \frac{2}{d+2}\|D_0\|_F^2. \tag{5}$$

Consequently,

$$\mathbb{E}_R\left[\min_{H\ diagonal}\|RD_0R^\top - H\|_F^2\right] = \frac{d}{d+2}\|D_0\|_F^2. \tag{6}$$

The proof is a short spherical moment calculation and is deferred to Appendix A. Intuitively, a diagonal entry of $RD_0R^\top$ is a random weighted average of the diagonal entries of D_0. Since D_0 is trace-free, the scalar component cancels, and only an $O(1/d)$ fraction of the energy remains visible on the exposed diagonal.

The trace-free condition is important. For a general diagonal edit $D = \mu I + D_0$, the scalar component is invariant: $R(\mu I)R^\top = \mu I$. Thus secret coordinates do not hide global rescalings; they hide feature-specific diagonal structure. The theorem should also not be read as a rank statement: $\mathrm{rank}(RDR^\top) = \mathrm{rank}(D)$. A sufficiently expressive dense low-rank adversary can therefore represent the hidden edit once its rank budget is high enough. Setting A proves a precise separation against exposed-basis diagonal or IA^3-style edits, not against arbitrary dense updates. The expected experimental pattern is canonical diagonal $\approx$ keyed diagonal $\gg$ naive exposed diagonal, with dense low-rank attacks recovering as the rank budget approaches $\mathrm{rank}(D)$.

4.2 From diagonal locality to block/head locality

Setting A shows that diagonal locality is basis-dependent: a feature-wise edit that is simple in the canonical hidden basis becomes dense after a secret change of basis. Setting B applies the same mechanism to a more model-relevant notion of locality. Instead of individual hidden coordinates, we consider heads, blocks, channel groups, or modules. Let the hidden space decompose as $\mathbb{R}^d = V_1 \oplus \cdots \oplus V_H$. A canonical block-local edit has the form $\Delta = \mathrm{blockdiag}(\Delta_1, \ldots, \Delta_H)$, and we write

$$\mathcal{E}_{\mathrm{block}} = \{\Delta : \Delta = \mathrm{blockdiag}(\Delta_1, \ldots, \Delta_H)\}.$$

A secret block-mixing basis sends this edit to $R\Delta R^\top$. Thus the authorised family is $\mathcal{A}_{k,\mathrm{block}} = \{R\Delta R^\top : \Delta \in \mathcal{E}_{\mathrm{block}}\}$, while the naive head-local adversary uses $\mathcal{E}_{\mathrm{block}}$ directly in the released basis. In words, the authorised editor knows the decomposition in which the edit is head-local; the naive adversary edits the wrong heads. For H equal blocks of size $q = d/H$, the block-local subspace has dimension $\dim(\mathcal{E}_{\mathrm{block}}) = Hq^2 = d^2/H$, whereas the full matrix space has dimension d^2. This suggests the dimension-predicted capture

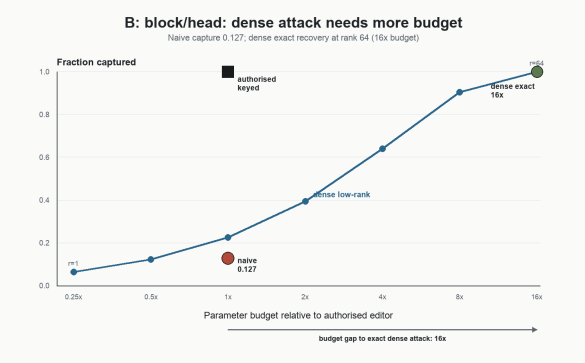
$$\frac{\|\Pi_{\mathrm{block}}(R\Delta R^\top)\|_F^2}{\|\Delta\|_F^2} \approx \frac{1}{H} \tag{7}$$

for generic block-local edits and sufficiently mixing R. The role of Setting B is therefore to test whether the diagonal-locality mechanism persists when locality means head- or block-locality rather than coordinate-wise diagonality. We test this prediction in Fig. 2a: at the authorised block-local budget, the naive exposed-basis head-local adversary captures little, while denser adversaries improve as their budget grows.

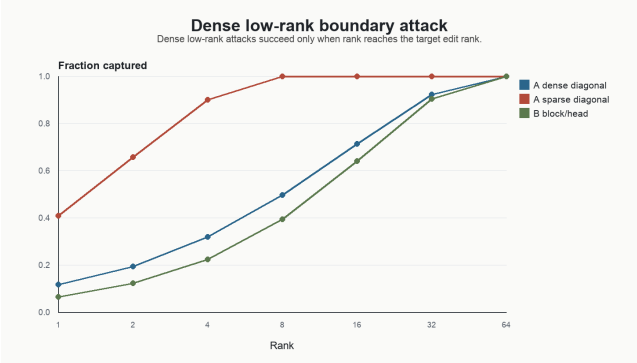
This is motivated by transformers: attention layers have exact basis-change subspace symmetries,

$$(W_Q, W_K, W_V, W_O) \mapsto (R_{QK}W_Q, R_{QK}^{-\top}W_K, W_V R_{VO}, R_{VO}^{-1}W_O)$$

preserves the QK logits and the value-output pathway. For real multi-head attention, the exact symmetry class must be chosen with care: within-head rotations, head permutations, and valid value/output changes are safer than arbitrary cross-head mixing, since different heads generally have different attention patterns.



(a) Setting B budget frontier.



(b) Dense low-rank boundary attack.

Figure 2: Budget frontier and boundary attack. Left: in Setting B, naive head-local edits at the authorised budget capture little, while denser attacks recover as budget increases. Right: dense low-rank attacks succeed once rank reaches the intrinsic rank of the hidden update, illustrating that secret coordinates break locality rather than rank.

Table 1: Edit-family comparison for Settings A and B. The authorised editor keeps locality because it knows the hidden basis. The naive adversary uses the same type of local edit in the released basis, where the target update is delocalised.

Setting	Authorised family	Naive exposed family	Boundary attack
A: dense diagonal	$RDR^\top$	exposed diagonal	dense rank-r, $r \geq \text{rank}(D)$
A: sparse diagonal	$RD_SR^\top$	exposed sparse diagonal	dense rank-$\lvert S\rvert$
B: block/head	$R\Delta R^\top$	exposed block-local	dense cross-block / sufficient rank

4.3 Comparison across edit families

The same-budget comparison across Settings A and B is shown in Fig. 1. The authorised editor always uses the local edit family in the keyed basis, so its capture is 1. The naive adversary uses the corresponding local family in the released basis. In Setting A, this means exposed diagonal or sparse-diagonal edits; in Setting B, this means exposed block/head-local edits. The resulting capture values are small: the naive edit has the right parameter count, but the wrong coordinate system. Table 1 summarises the expected behaviour. The positive result is strongest for edit families whose efficiency depends on coordinate locality. The boundary attack is dense low-rank editing: once the adversary is allowed a sufficiently expressive dense update, the hidden edit can be represented.

If a layer is reparameterised by an invertible basis change and $\Delta W = UV^\top$, then the transformed update has the same rank: $A\Delta W B^{-1} = (AU)(B^{-\top}V)^\top$. Therefore secret coordinates cannot generically force an unrestricted rank-r update to become higher-rank. Fig. 2b shows this boundary directly: dense low-rank attacks recover once their rank reaches the intrinsic rank of the hidden target update. What secret coordinates can do is make local edit families insufficient, forcing the adversary to replace locality with density, rank, compute, or gauge recovery.

4.4 Optimiser-bounded locality

Settings A and B are representational: the unauthorised local edit family does not contain the hidden edit except through projection. Setting C studies a different failure mode. Here the edit may be

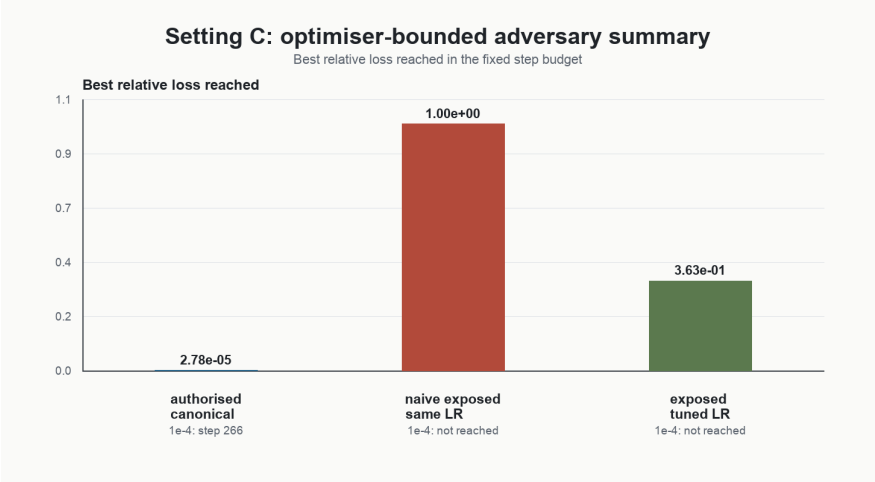


Figure 3: Optimiser-bounded behaviour in Setting C. The non-orthogonal release preserves the function and leaves the target representable, but exposed-coordinate optimisation can require more tuning or more steps than authorised canonical optimisation.

representable in the released coordinates, but optimisation in those coordinates becomes difficult. The simplest example is a non-orthogonal hidden-basis symmetry: for an invertible matrix S, release $A' = SA$ and $B' = BS^{-1}$. Correctness is unchanged, since $B'A' = BS^{-1}SA = BA$. However, if S is ill-conditioned, the exposed parameterisation can have badly scaled gradients and a poorly conditioned Hessian, even though the underlying function is identical.

The adversary's budget is therefore not only a parameter or rank budget, but an optimisation budget such as $B = (N_{\text{samples}}, T_{\text{steps}}, \text{optimizer}, \text{precision}, \eta)$. A naive or aware-but-unkeyed adversary may use an edit family expressive enough to represent the target update, but still fail to reach the task threshold within the allowed steps, data, precision, or learning-rate range. This setting is especially relevant for first-order optimisers, quantised training, fixed PEFT recipes, and low-data adaptation.

Thus Setting C extends the locality story from representation to optimisation. Orthogonal secret coordinates can hide the basis in which an edit is local; non-orthogonal secret coordinates can also make the exposed coordinates unfriendly to standard optimisation. The claim is not that the edit is impossible to represent, but that the adversary's training recipe may need more tuning, preconditioning, data, precision, or steps. Fig. 3 illustrates this optimiser-bounded effect: the target update remains representable after the non-orthogonal release, but exposed-coordinate optimisation is substantially less effective under the same training budget.

Table 2 summarises the three settings. Settings A and B create a cost gap because the naive adversary uses the right type of local edit in the wrong coordinates. Setting C preserves representability but can increase optimisation cost under a fixed training budget.

5 Discussion and Conclusion

Limitations. We show that a key can instill coordinate locality: in Settings A and B, the authorised editor uses the local edit family in the hidden basis, while the naive adversary uses the same nominal family in the released basis and therefore captures much less of the target update. To recover, the adversary must move to a denser family, increase rank, recover the gauge,

Table 2: Cost-separation summary. Settings A and B break coordinate locality; Setting C preserves representability but can increase optimisation cost. Dense recovery costs are absolute parameter counts, not multipliers.

Setting	Auth. cost	Naive cost	Naive capture	Recovery route
A: diagonal	d	d	$2/(d+2)$	dense rank-d, $\approx 2d^2$ params
A: s-sparse diag.	s	s	small	dense rank-s, $\approx 2ds$ params
B: block/head	d^2/H	d^2/H	$\approx 1/H$	dense cross-block, $\approx d^2$ params
C: optimiser	same repr.	same repr.	opt.-dependent	preconditioning, tuning, or more steps

or spend more optimisation budget. The central boundary is rank invariance: for any invertible basis change $W' = AWB^{-1}$ and any LoRA update $\Delta W = UV^\top$, the transformed update is $A\Delta WB^{-1} = (AU)(B^{-\top}V)^\top$, so $\mathrm{rank}(A\Delta WB^{-1}) = \mathrm{rank}(\Delta W)$. Thus secret coordinates do not make open-weight models secure in general: they do not prevent sufficiently expressive dense adapters, full fine-tuning, canonicalisation, prompt attacks, activation steering, distillation, or key leakage. Instead, for named coordinate-local edit families, keyed symmetries can create edit-cost separation. Limiting attacks are therefore dense or equivariant adapters, gauge recovery, function-space attacks, optimisation attacks such as preconditioning or rescaling, and failures of key management.

Future directions. A next step is to move from clean locality models to architecture-aware secret coordinates, by identifying which symmetry groups are available in real transformer components. A second direction is adversarial key design and cryptanalysis: choose keys that minimise projection onto known edit families, then test whether canonicalisation, equivariant attackers, or adapter transfer can recover the hidden gauge.

Together, these directions would move secret coordinates from a geometric proof-of-concept toward a cryptographic primitive for open-weight models. The goal is not to encrypt inference, but to define a keyed access structure over editability: authorised users know a trapdoor coordinate system in which certain edits are cheap, while unkeyed users face a higher-cost edit-forgery problem. This reframes parameter symmetries as more than non-identifiability: they become a source of keyed hardness assumptions about the geometry of model modification. Secret coordinates do not hide what the model computes; they shape who can cheaply change it.

References

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=CQsmMYmlP5T. 2, 3

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115, 2023. 1

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International conference on machine learning*, pages 3259–3269. PMLR, 2020. 2

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee's mergekit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, 2024. 1

Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. On the symmetries of deep learning models and their internal representations. *Advances in Neural Information Processing Systems*, 35:11893–11905, 2022. 1, 2

Robert Hecht-Nielsen. On the algebraic structure of feedforward network weight spaces. In *Advanced Neural Computers*, pages 129–135. Elsevier, 1990. 1, 2

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe loRA: The silver lining of reducing safety risks when finetuning large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=HcifdQZFZV. 1

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. URL https://openreview.net/forum?id=nZeVKeeFYf9. 1

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj. 1

Derek Lim. *Symmetries in Neural Network Functions and Parameters*. PhD thesis, Massachusetts Institute of Technology, 2025. 1, 2

Derek Lim, Theo Putterman, Robin Walters, Haggai Maron, and Stefanie Jegelka. The empirical impact of neural parameter symmetries, or lack thereof. *Advances in Neural Information Processing Systems*, 37:28322–28358, 2024. 1, 2, 3

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin A Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35:1950–1965, 2022. 1

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024. 1

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022. 1

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in neural information processing systems*, 32, 2019. 2

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, volume 2024, pages 30988–31043, 2024. 1

Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, Andy Zou, Dawn Song, Bo Li, Dan Hendrycks, and Mantas Mazeika. Tamper-resistant safeguards for open-weight LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FIjRodbW6. 1

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR, 2022. 1

A Proof of Theorem 4.1

Let $D_0 = \mathrm{diag}(s_1, \dots, s_d)$ with $\sum_a s_a = 0$, and let $R \sim \mathrm{Haar}(O(d))$. For a fixed diagonal entry,

$$(RD_0R^\top)_{ii} = \sum_{a=1}^{d} s_a R_{ia}^2.$$

A row of a Haar orthogonal matrix is distributed as $x \sim \mathrm{Unif}(S^{d-1})$. Hence

$$\mathbb{E}\big[(RD_0R^\top)_{ii}^2\big] = \mathbb{E}\left[\left(\sum_a s_a x_a^2\right)^2\right].$$

Using $\mathbb{E}[x_a^4] = 3/(d(d+2))$ and $\mathbb{E}[x_a^2 x_b^2] = 1/(d(d+2))$ for $a \neq b$,

$$\mathbb{E}\left[\left(\sum_a s_a x_a^2\right)^2\right] = \frac{3}{d(d+2)}\sum_a s_a^2 + \frac{1}{d(d+2)}\sum_{a\neq b} s_a s_b.$$

Since $\sum_a s_a = 0$, we have $\sum_{a\neq b} s_a s_b = -\sum_a s_a^2$. Therefore

$$\mathbb{E}\big[(RD_0R^\top)_{ii}^2\big] = \frac{2}{d(d+2)}\|D_0\|_F^2.$$

Summing over $i = 1, \dots, d$ gives

$$\mathbb{E}_R\left[\|\Pi_{\mathrm{diag}}(RD_0R^\top)\|_F^2\right] = \frac{2}{d+2}\|D_0\|_F^2.$$

Finally, the best diagonal approximation to any matrix M in Frobenius norm is $\Pi_{\mathrm{diag}}(M)$. Since orthogonal conjugation preserves Frobenius norm,

$$\min_{H \text{ diagonal}} \|RD_0R^\top - H\|_F^2 = \|D_0\|_F^2 - \|\Pi_{\mathrm{diag}}(RD_0R^\top)\|_F^2.$$

Taking expectations proves the residual identity.

B Loss as a weighted Frobenius projection

Let $z' = A'x$, and suppose the target exposed-coordinate update is $M_\star$. An edited released model with hidden update M computes $f_M(x) = B'(I + M)A'x$. The population squared loss against the target edit is

$$\mathcal{L}(M) = \mathbb{E}_x\left[\|B'(M - M_\star)z'\|_2^2\right].$$

Writing $G_B = B'^\top B'$ and $\Sigma_{z'} = \mathbb{E}[z'z'^\top]$, this becomes

$$\mathcal{L}(M) = \mathrm{tr}\left(G_B(M - M_\star)\Sigma_{z'}(M - M_\star)^\top\right).$$

In the clean isotropic case $G_B = I$ and $\Sigma_{z'} = I$, so $\mathcal{L}(M) = \|M - M_\star\|_F^2$. Thus the best adversarial update in a linear edit family is exactly the Frobenius projection of $M_\star$ onto that family. In the general case, the same statement holds with a weighted Frobenius norm.

C Additional projection and budget checks

Figure 4 verifies that the sampled-rotation projection calculations match the predicted capture values for the diagonal and block/head settings. Figures 5a and 5b show the detailed Setting A budget frontiers. They are included for completeness; the main text uses the aggregate same-budget comparison and dense low-rank boundary plot.

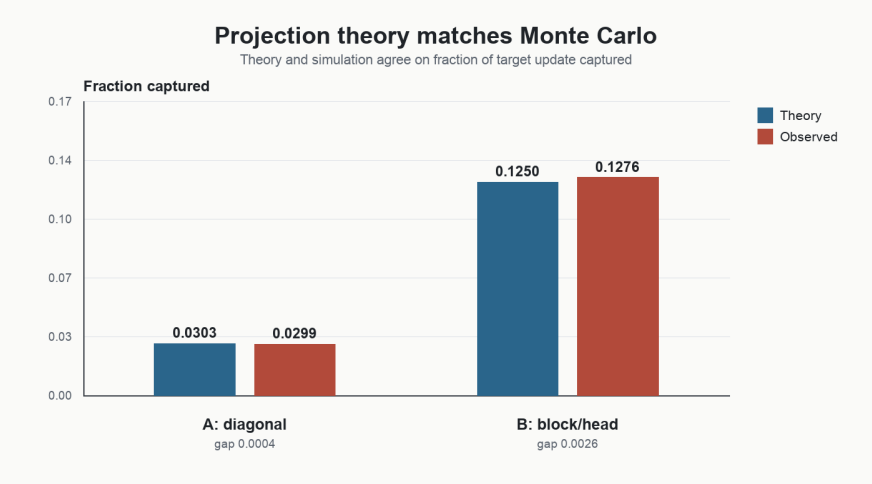
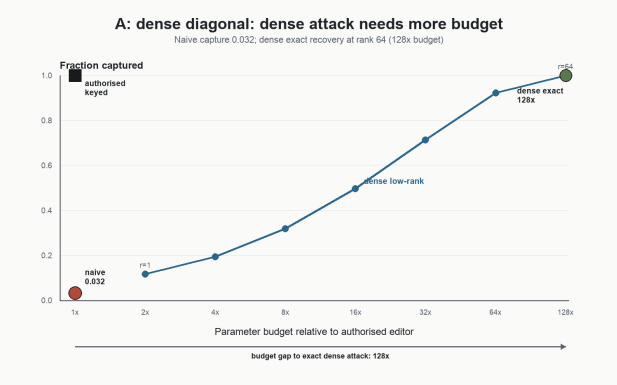
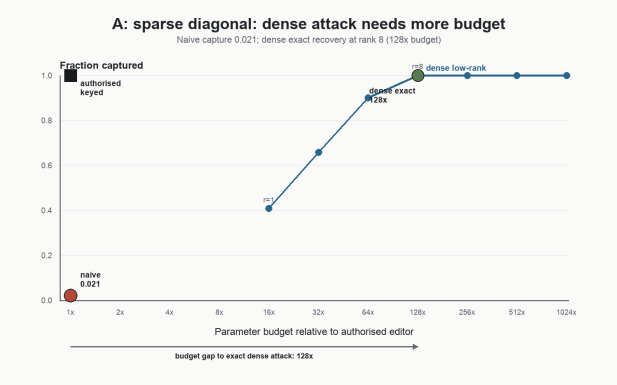


Figure 4: Projection sanity check. The predicted capture values agree with sampled random-basis measurements for the diagonal and block/head settings.



(a) Dense diagonal: naive diagonal edits fail at matched budget; dense low-rank attacks recover with larger rank.



(b) Sparse diagonal: keyed sparse edits are cheap; exposed edits require dense low-rank recovery.

Figure 5: Setting A budget frontiers for dense and sparse diagonal targets.