Finite Transfer Audits in Presentation-Langlands

Luca Blanchi

June 2026

Abstract

Functorial transfers are usually global mathematical objects, but many computational uses of them take place inside finite windows. This note gives a presentation-theoretic audit format for such finite transfer problems. A transfer computation should return not only a proposed match, but one of three finite-window states: MATCH, NO_TARGET, or AMBIGUOUS, each with the observables, budget, normalization conventions, and residual ledger that justify it.

The main theorem is a finite-window transfer audit. Given a finite source window, a finite target window, explicit transfer prediction rules on a separating target fingerprint, and a declared ramification ledger, the audit is sound in the window: a unique match is a unique target in the declared window; no match proves no target in that window; unresolved local data or incomplete target coverage must be returned as ambiguity. The theorem is formal, but it becomes useful in arithmetic settings where recent explicit formulae or local converse theorems supply the prediction rules.

The note develops three modules: base-change audits, ramification ledgers controlled by local gamma or Bernstein observables, and packet-period resolvers for groups where Hecke data identify a parameter or packet rather than an individual representation. The common pattern is observable substitution: one family of observables reduces the problem to a finite residual fibre, and a second family is chosen to resolve that fibre.

1 Introduction

Langlands functoriality relates objects through transfers: base change, automorphic induction, endoscopic transfer, theta lifts, symmetric powers, tensor products, and many other constructions. Computationally, however, one often has only finite data: a bounded source list, a bounded target list, a set of local formulae, and a range of Hecke values or local invariants.

This note treats such a computation as an audit problem. The output should not be a bare assertion that a match has been found. It should be a finite record explaining:

(i) the declared source and target windows;

(ii) the transfer rules used to predict target observables;

(iii) the target fingerprint that separates the declared target window;

(iv) the places where ramification or normalization creates unresolved data;

(v) whether the result is a match, a non-existence statement in the window, or an ambiguity.

The central design principle is trichotomy:

$$\text{MATCH}, \qquad \text{NO\_TARGET}, \qquad \text{AMBIGUOUS}.$$

Collapsing these cases is the main failure mode of informal computational functoriality.

2 Finite transfer packages

Definition 2.1 (Finite transfer package)**.** *A finite transfer package is a tuple*

$$\mathsf{T} = (X, Y, \mathcal{O}_Y, J, \mathcal{R}, \mathsf{N}, \mathsf{RamLed}, \mathsf{V}),$$

where X is a finite source window, Y is a finite target window, $\mathcal{O}_Y$ is a family of target observables, $J \subseteq \mathcal{O}_Y$ is a selected target fingerprint, $\mathcal{R}$ is a set of prediction rules sending source data to predicted target values on J, N is the normalization ledger, RamLed is the ramification ledger, and V is the verification procedure.

The target window is part of the theorem, not background context. It must record the relevant weights, levels, characters, coefficient fields, embeddings, Galois conjugacy conventions, old/new decomposition, possible level drops, and ramified-prime restrictions. Without such a completeness ledger, the output NO_TARGET has no sound meaning.

The target fingerprint J separates Y if

$$y_1 \neq y_2 \quad \Longrightarrow \quad \exists q \in J \text{ such that } q(y_1) \neq q(y_2).$$

More generally, J may separate Y only modulo a declared equivalence relation. Then the residual ledger must state the remaining equivalence classes.

3 The finite transfer audit

For $x \in X$, suppose the prediction rules produce a partial vector

$$\mathrm{Pred}_J(x) = (p_q(x))_{q \in J_0}$$

on a subset $J_0 \subseteq J$. Missing entries may arise from ramified places, unavailable local rules, incompatible normalizations, or undeclared target levels.

Definition 3.1 (Audit output)**.** *The audit returns one of:*

$$\mathrm{MATCH}(x, y),$$

meaning a unique target $y \in Y$ matches all required predicted values;

$$\mathrm{NO\_TARGET}(x),$$

meaning no target in the declared target window matches the required predicted values;

$$\mathrm{AMBIGUOUS}(x, \mathsf{L}),$$

meaning the available data leave a residual ledger L, or that the target window or local prediction rules are incomplete.

Theorem 3.2 (Finite transfer audit)**.** *Let Y be a finite target window and let J be a fingerprint separating Y. Suppose the transfer prediction rules for $x \in X$ produce values $p_q(x)$ for every $q \in J$, in the same normalization as the target data.*

If there is a unique $y \in Y$ such that

$$q(y) = p_q(x) \quad \text{for every } q \in J,$$

then y is the unique target in the declared window.

If there is no such y, then there is no target in the declared window with the predicted fingerprint.

If some predicted values are unavailable, if J does not separate the relevant target window, or if the target window does not include all candidates allowed by the ramification ledger, then the sound output is AMBIGUOUS *with the corresponding residual ledger.*

Proof. The first statement follows from separation: two targets matching the same full J-fingerprint would be equal in the declared window. The second is just the definition of non-existence inside a finite declared target list. The third records the failure of one of the hypotheses needed for the first two conclusions. Missing prediction rules, non-separating fingerprints, or incomplete target coverage cannot justify either uniqueness or non-existence. □

The theorem is deliberately finite. It does not assert global functoriality. It gives the exact logical status of a finite transfer computation.

Thus NO_TARGET always means no target in the declared target window. It never means that a global transfer does not exist unless an independent theorem identifies the declared window with the complete global search space.

Example 3.3 (Three audit outcomes)**.** *Let $Y = \{y_1, y_2, y_3\}$ be a declared target window and let $J = \{q_1, q_2\}$ separate it. If the predicted vector for x equals the J-fingerprint of y_2 and no other target, the output is* MATCH(x, y_2)*. If it equals no fingerprint in Y, the output is* NO_TARGET(x)*, meaning no target in this finite Y. If the value of q_2 is unavailable, or if an omitted ramified case could add a fourth target y_4, the sound output is* AMBIGUOUS*, not* MATCH *or* NO_TARGET*.*

4 Base-change audit

Let f be a classical newform and let $F/\mathbb{Q}$ be a number field in a class where explicit base-change formulae are available. Let $\mathcal{H}$ be a finite Hilbert modular target window: possible weights, levels, characters, and coefficient fields in the declared range.

The target-completeness ledger for $\mathcal{H}$ must include old/new decomposition, Galois conjugacy, coefficient-field embeddings, possible level drops, ramified primes, and the local conventions used at primes above the level. These entries are what make a finite NO_TARGET statement auditable.

Let J be a separating Hecke fingerprint for $\mathcal{H}$. For $\mathfrak{P} \in J$, explicit base-change rules predict values

$$\mathrm{Pred}_{\mathfrak{P}}(f, F).$$

Proposition 4.1 (Verified base-change audit)**.** *Assume that J separates the declared Hilbert target window $\mathcal{H}$, and that all predicted values on J are defined in the recorded normalization. Then the finite transfer audit applied to these data returns a sound* MATCH *or* NO_TARGET *statement inside $\mathcal{H}$. If ramified primes, level changes, or target completeness are unresolved, the sound output is* AMBIGUOUS *with a ramification ledger.*

Proof. This is the finite transfer audit theorem applied to the base-change prediction rules. Ramified primes and target completeness are exactly the hypotheses required to know that all candidates and all predicted fingerprint values have been included. □

The base-change audit is a model case because recent work provides explicit formulae and computational procedures for base-change lifts and Hida-family lifts in concrete settings [1]. Presentation Theory adds the audit discipline: a proposed lift is accompanied by the finite target window, the fingerprint, the predicted values, the ramification ledger, and the reason the output is one of the three possible states.

5 Ramification ledgers

Unramified data are often cheap to compare. The difficult part of a transfer computation is the ramified local data. We package this difficulty as a ledger.

Definition 5.1 (Local ramification ledger)**.** *At a bad place v, let $\mathcal{L}_v$ be the finite set of local candidates compatible with the currently observed global data. A local ramification ledger is*

$$\mathsf{RamLed}_v = (\mathcal{L}_v, \mathcal{Q}_v, c_v, F_v),$$

where $\mathcal{Q}_v$ is a family of local observables, c_v records their cost, and F_v records the residual fibre after the selected local tests.

Possible local observables include conductors, depths, types, Bernstein components, local L-factors, epsilon factors, and gamma factors

$$\gamma(s, \pi_v \times \tau, \psi).$$

Proposition 5.2 (Local resolver)**.** *Let $\mathcal{L}_v$ be finite. If a family of local observables separates $\mathcal{L}_v$, then some finite subfamily separates $\mathcal{L}_v$. A minimum-cost separating subfamily is a weighted set-cover problem on pairs of local candidates.*

Proof. For each pair of distinct local candidates, choose an observable that separates them. Finiteness gives a finite separating set. The exact optimization formulation is weighted set cover: each observable covers the pairs it separates. □

Local converse theorems provide the arithmetic input that certain classes of gamma-factor observables separate representations. The finite proposition then turns that input into a concrete local audit in any bounded computational window.

6 Packet and period resolvers

For groups beyond GL_n, Hecke data may identify an Arthur or Langlands parameter, or a packet, rather than an individual automorphic representation. A correct record must distinguish these two levels.

Let Π_ψ be a finite packet in a declared window. Define the packet ledger

$$\mathsf{PktLed}(\pi) = \{\pi' \in \Pi_\psi : \mathcal{O}_{\mathrm{Hecke}}(\pi') = \mathcal{O}_{\mathrm{Hecke}}(\pi)\}.$$

Period observables, local signs, component-group characters, and branching data may resolve this ledger.

Proposition 6.1 (Packet resolver in a finite window)**.** *Let* PktLed *be finite. If period or local-sign observables separate the members of* PktLed*, then a finite subset of these observables separates the ledger. Choosing a minimum-cost resolver is weighted set cover on pairs in the packet ledger.*

Proof. The proof is the same finite separation argument. Each period or local-sign observable covers the pairs of packet members on which it takes different values. A list resolves the packet precisely when the covered pairs are all pairs. □

This module is not a theorem that periods always resolve packets cheaply. It is a clean way to state and test such claims:

$$\text{Hecke parameter} \longrightarrow \text{packet ledger} \longrightarrow \text{period or local-sign resolver.}$$

The arithmetic content of this module lies in proving that natural periods or local signs separate natural packets with small cost. The finite set-cover statement only records the consequence once such separating observables are supplied.

7 Composition of audits

Suppose

$$X \xrightarrow{F} Y \xrightarrow{G} Z$$

are finite transfer packages. Each has a target fingerprint, a normalization ledger, and a residual fibre. Their composition is valid only if the normalization output of F matches the normalization input of G, and if the unresolved ledger of F is either resolved before applying G or transported as part of the composed ledger.

An ambiguity left by the first audit cannot be ignored by the second. It must either be killed by additional observables on Y, or carried forward into the composed ledger on Z.

Proposition 7.1 (Composed audit)**.** *If two finite transfer audits satisfy compatible normalizations and if their residual ledgers are transported explicitly, then the composed audit has residual ledger contained in the fibre product of the two ledgers over the intermediate window.*

Proof. A candidate for the composed transfer must be compatible with the residual possibilities left by the first audit and with those allowed by the second. Thus its ambiguity lies in the common refinement of the two residual fibres, which is their fibre product over the intermediate data. □

This is the finite counterpart of controlled transfer in Presentation Theory. The overhead is not only a numeric cost; it includes normalizations, local unresolved data, and fibre growth.

8 Conjectural cost questions

The finite audit theorem is formal. The mathematical content enters through cost bounds for natural resolvers. The following questions are intended as falsifiable research targets.

(i) In bounded conductor and depth families for $\mathrm{GL}_n(F_v)$, do gamma-factor and Bernstein observables resolve local ramification ledgers with polynomial cost in n, conductor, depth, and $\log q_v$?

(ii) In natural packet families for classical groups, do a small number of period or local-sign observables resolve most packet ledgers?

(iii) In explicit base-change computations, how large is the target fingerprint needed before ambiguous ramified cases disappear in practice?

These questions are not part of the formal theorem. They are the arithmetic problems exposed by the audit format.

References

[1] I. Blanco-Chacón, L. Dieulefait, and A. Haavikko, *On the computation of base-change lifts and lifts of Hida families*, arXiv:2604.05618, 2026.

[2] J. E. Cremona, *Hecke operators, Hecke eigensystems, and formal modular symbols over number fields*, arXiv:2601.17524, 2026.

[3] H. Jacquet, I. Piatetski-Shapiro, and J. Shalika, *Rankin-Selberg convolutions*, American Journal of Mathematics 105, 1983.

[4] W. T. Gan, B. Gross, and D. Prasad, *Symplectic local root numbers, central critical L values, and restriction problems in the representation theory of classical groups*, Asterisque 346, 2012.