A Spectral Theory of Computation

Charles Renshaw-Whitman
MATS 9.0 Final Report

March 2026

Draft — March 2026. Comments and corrections welcome.

Abstract

Any computation that passes through a hidden state — such as a neural network split at an intermediate layer — can be decomposed into an encoder (input to hidden) and a decoder (hidden to output). We develop a spectral theory for this decomposition using kernel canonical correlation analysis, which extracts independent *modes*: channels of information flow from input to output through the hidden state. Each mode has an encoder strength (how much input information it carries), a decoder strength (how well it predicts the output), and a routing weight (how the encoder mode connects to the decoder mode through the hidden state). The central object is the *routing matrix* R, whose entries R_{jk} give the overlap between encoder and decoder modes in the hidden-state RKHS. The routing matrix is gauge-invariant: it depends only on the immediate inputs and outputs of the hidden state, not on how the hidden state is parameterized or how the surrounding computation is organized. We prove that the routing data (σ, τ, R) is the complete invariant of a factored computation up to reparameterization, that truncating to the top modes is optimal in a precise sense, that information-routing strengths can only decay with depth, and that estimation requires a number of samples controlled by the spectral gap. The output in the decomposition is a free parameter: replacing it with any behavioral measure (e.g., a truthfulness probe, a toxicity score) gives a behavior-specific routing matrix, and these compose under post-processing.

Executive Summary

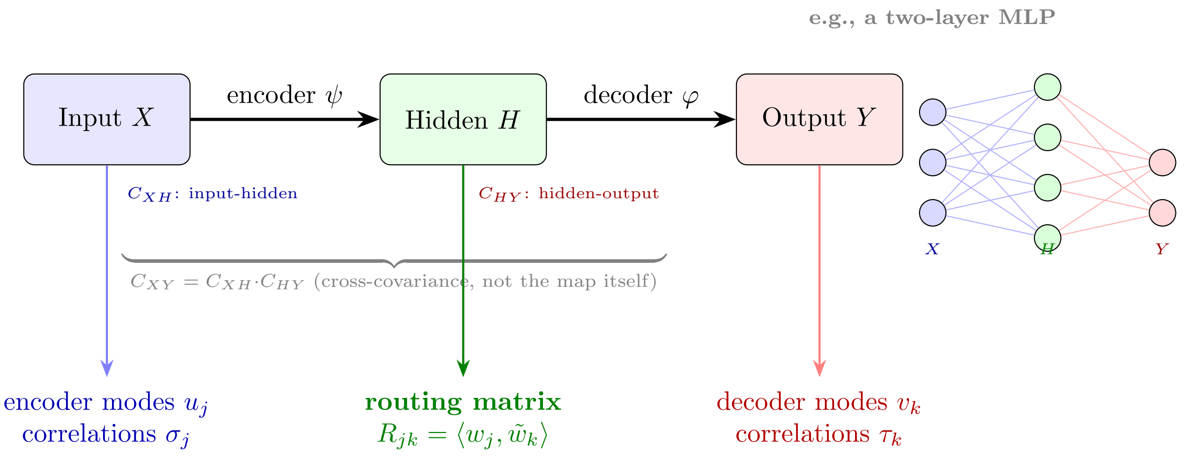
This summary covers the key ideas in two pages. The full report follows.

Two questions

1. **What do the hidden-state activations mean?** Not which features are linearly decodable (probing answers that), but how each component of the hidden state connects to specific downstream behavior.
2. **How can I tell if a network is implementing a given algorithm?** Given a white-box but illegible neural network, how do we compare its internal organization to a reference computation?

Both questions are relevant to alignment: if we could decompose a model's hidden states into interpretable channels, that would help evaluate whether its behavior is safe; and if we could compare internal computations, that would help verify whether safety properties transfer across models or training runs. The routing matrix is a step toward this goal, though it has not yet been tested on the models where these questions are most urgent.

The **routing matrix** addresses both by decomposing any factored computation $X \to H \to Y$ into independent *modes* with measurable strengths.



$$\widetilde{C}_{XY} = \sum_{j,k} \sigma_j \, R_{jk} \, \tau_k \, u_j \otimes v_k$$

Figure 1: Routing decomposition of a factored computation $X \to H \to Y$. Encoder and decoder modes are extracted via CCA; the routing matrix R_{jk} captures how encoder modes connect to decoder modes through the hidden state.

- σ_j measures how much information encoder mode j carries from the input
- τ_k measures how well decoder mode k predicts the output
- R_{jk} measures how mode j is *wired* to mode k through the hidden state

- The triple $(\boldsymbol{\sigma}, R, \boldsymbol{\tau})$ is **gauge-invariant**

In this context, *gauge-invariant* means the routing matrix is unchanged if you rotate, permute, or apply any invertible linear transform to the neurons in the hidden layer (the next layer's weights compensate, so the network computes the same function). More generally, it is invariant to how the prior computation is organized — it depends only on the statistical relationship between the hidden state's immediate inputs and outputs, not on the coordinate system used to describe the hidden state or the structure of the computation that produced it. This is what makes it possible to compare two networks that use completely different internal representations.

Observable substitution: behavior-specific routing

The output Y is a free parameter. By choosing different behavioral targets, the *same* hidden state yields different routing matrices:

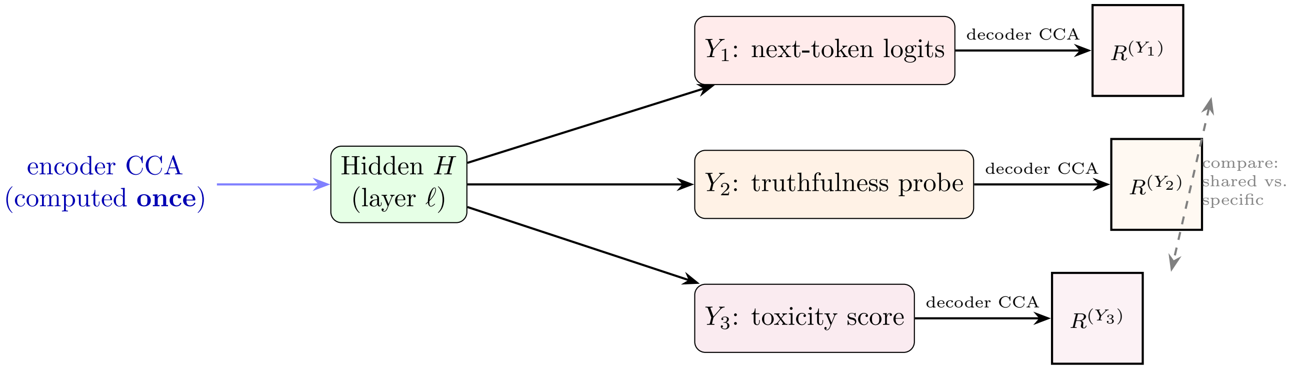


Figure 2: Observable substitution: the same hidden state yields different routing matrices for different behavioral targets. The encoder CCA is computed once; each new observable requires only a decoder CCA.

The encoder decomposition is computed **once per layer**; each new behavioral question requires only a new decoder CCA. Post-processing an observable **composes**: the routing for a transformed output (e.g., applying a probe to logits) can be derived from the original routing without redoing the encoder step.

Comparing two systems

Given two systems with the same I/O spaces, the comparison decomposes into three levels — each answering a different question:

Results

1. **The routing matrix is a complete invariant.** Two factored computations $X \to H \to Y$ have the same routing data $(\boldsymbol{\sigma}, R, \boldsymbol{\tau})$ if and only if they differ only by a reparameterization of the hidden state. The routing matrix captures everything about the computation that is independent of the choice of coordinates on H.

2. **Truncating to the top modes is optimal.** If you want the best k-mode approximation to a factored computation — keeping k_1 encoder modes and k_2 decoder modes — the SVD-

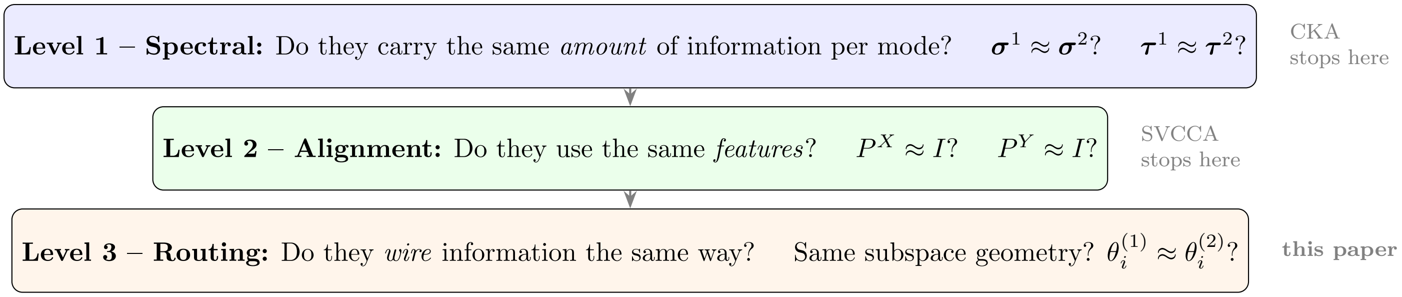

Level 1 – Spectral: Do they carry the same *amount* of information per mode? $\boldsymbol{\sigma}^1 \approx \boldsymbol{\sigma}^2$? $\boldsymbol{\tau}^1 \approx \boldsymbol{\tau}^2$?
CKA stops here
Level 2 – Alignment: Do they use the same *features*? $P^X \approx I$? $P^Y \approx I$?
SVCCA stops here
Level 3 – Routing: Do they *wire* information the same way? Same subspace geometry? $\theta_i^{(1)} \approx \theta_i^{(2)}$?
this paper

Figure 3: Three levels of system comparison: spectral (do they carry the same amount of information?), alignment (do they use the same features?), and routing (do they wire information the same way?). Existing methods stop at level 1 or 2; this paper addresses level 3.

based truncation is provably optimal (in Hilbert-Schmidt norm). The approximation error is controlled by the discarded singular values.

3. **Information-routing strengths decay with depth.** For a computation that passes through multiple intermediate states $X \to H_1 \to H_2 \to Y$, the unwhitened cross-covariance singular values can only decrease at each step. All structural change is absorbed by the routing matrix: the canonical correlations are invariant under deterministic transitions, while the routing matrix records how modes are rearranged.

4. **Estimation requires samples controlled by the spectral gap.** To reliably resolve the top k modes, you need $N \gg k^2/\text{gap}^2$ samples, where the gap is between adjacent canonical correlations. Closely spaced modes are individually unreliable, but aggregate properties (total spectral mass, block structure of R) remain robust.

5. **Behavioral observables compose.** Post-processing the output (e.g., applying a probe to logits) yields a new routing matrix that can be computed from the original without redoing the encoder decomposition. This means the encoder is reusable across behavioral questions.

Status and next steps

This is a *theoretical* contribution: a formal framework for decomposing information flow through hidden states. It is validated on toy systems (linear maps, MLPs, finite-state transducers); whether kernel CCA modes correspond to semantically meaningful features in transformer language models — where skip connections violate the Markov assumption at every layer — is the central open empirical question. The framework's value for alignment is in making interpretability claims *precise*: it gives a formal language for statements like "this hidden state contributes to this behavior via these modes," complete with estimation guarantees and gauge invariance.

Contents

1 Introduction

1.1 The problem

At any intermediate point in a computation we can split it into an *encoder* $\psi\colon X \to H$ and a *decoder* $\varphi\colon H \to Y$. The end-to-end map $f = \varphi \circ \psi$ is the same regardless of where we slice, but the hidden state H carries information about *how* the computation is organized.

More precisely, comparing two computations through their hidden states requires answering three questions that existing tools conflate:

(a) **Do they use the same features?** Do the hidden representations pick out the same directions in input space?

(b) **Do they carry the same amount of information?** Is the spectrum of correlations between input and hidden state similar?

(c) **Do they wire information the same way?** If both systems detect feature j in the input, do they route it to the same output mode k?

A key theoretical feature is *observable substitution*: the output Y in the decomposition is a free parameter. Replacing it with different behavioral measures (next-token prediction, a truthfulness probe, a toxicity score) gives different routing matrices for the same hidden state, formalizing the notion of behavior-specific information flow. Observable substitution composes (Theorem 4.3): post-processing an observable yields a routing matrix derivable from the original.

Existing tools (CKA, SVCCA, probing, activation patching) answer these questions only partially; we compare in detail in §3.6.

1.2 What this report delivers

We develop a spectral theory for factored computations, built on kernel canonical correlation analysis (CCA). The central object is the **routing matrix** R, which decomposes the information flow through a hidden state into independent modes.

The main contributions are:

1. A **gauge-invariant factorization signature**: the routing matrix R_{jk} captures how input modes connect to output modes through the hidden state, invariant under reparameterization of the hidden space (§3).
2. A **three-level decomposition** of representation comparison into spectral, mode-alignment, and routing components, refining CKA into interpretable pieces (§4.3).
3. **Optimality guarantees**: Eckart-Young theory for factored operators shows that the SVD-based decomposition is optimal for low-rank approximation, and that the routing matrix determines which modes matter for the end-to-end computation (§5).
4. A **spectral data processing inequality** for temporal slices of a computation, showing that unwhitened cross-covariance singular values decay with depth while canonical correlations are invariant, with all structural change absorbed by the routing matrix (§5.2).
5. **Estimation error bounds** that determine how many samples are needed to reliably estimate the routing matrix, and when the spectral gaps are too small to resolve (§5.3).
6. **Observable substitution**: the output Y can be replaced by any behavioral measure, giving routing matrices specific to particular behaviors (§4).

1.3 Notation

$X = \mathbb{R}^{d_X}$, $H = \mathbb{R}^{d_H}$, $Y = \mathbb{R}^{d_Y}$ are the input, hidden, and output spaces. Kernels k_X, k_H, k_Y on these spaces generate RKHSs $\mathcal{H}_X, \mathcal{H}_H, \mathcal{H}_Y$. We write $\langle \cdot, \cdot \rangle_{\mathcal{H}_H}$ for the RKHS inner product and $\| \cdot \|_{\mathrm{HS}}$ for the Hilbert-Schmidt norm on operators.

2 Background

2.1 Why not L^2: the isometry obstruction

Before developing the framework, we explain why the naive approach fails, motivating the use of reproducing kernel Hilbert spaces.

A first attempt is to study the *Koopman operator* $K_\psi \colon L^2(H, \mu_H) \to L^2(X, \mu_X)$ defined by $K_\psi g = g \circ \psi$, where $\mu_H = \psi_* \mu_X$ is the pushforward measure. This operator is linear even when ψ is not, and its singular value decomposition would give a spectral comparison of the computation.

The problem is immediate:

Proposition 2.1 (Isometry obstruction)**.** *If $\mu_H = \psi_* \mu_X$, then $K_\psi \colon L^2(\mu_H) \to L^2(\mu_X)$ is an isometry. Every singular value equals* 1.

Proof. By the change-of-variables formula:

$$\|K_\psi g\|^2_{L^2(\mu_X)} = \int_X |g(\psi(x))|^2 \, d\mu_X(x) = \int_H |g(h)|^2 \, d(\psi_* \mu_X)(h) = \|g\|^2_{L^2(\mu_H)}.$$

Since K_ψ preserves norms, it is an isometry, and all singular values equal 1. □

This holds regardless of dimensions, smoothness, or structure of ψ. The space L^2 is “too large”: it contains functions sensitive to every measurable set, so it can perfectly track any deterministic map.

2.2 Why not mutual information?

An alternative is to use mutual information $I(X; H)$ to quantify the coupling between input and hidden state. But mutual information on continuous spaces is notoriously difficult to estimate: it requires density estimation or variational bounds (e.g., MINE; Belghazi et al. [25]), and even state-of-the-art estimators have high variance and are sensitive to hyperparameters, because MI is defined via density ratios, which are hard to estimate in high dimensions.

Kernelized versions of mutual information — such as the Hilbert-Schmidt Independence Criterion (HSIC; Gretton et al. [24]) or kernel mutual information (Bach [3]) — improve the situation by replacing density estimation with kernel evaluations. But once you kernelize, you are already working with cross-covariance operators between RKHSs: HSIC is the squared Hilbert-Schmidt norm of C_{XH}, and kernel MI is a function of the singular values of C_{XH}. Kernelizing gets you most of the way to our framework. The remaining step — decomposing the SVD of C_{XH} through the factored structure to extract the routing matrix — is what this report contributes.

2.3 The fix: restricted function spaces

Work with a *restricted* function space where not all functions are available. A reproducing kernel Hilbert space (RKHS) provides exactly this: the kernel determines which functions are in the space, and the choice of kernel controls which aspects of the map are visible. The cross-covariance operator on an RKHS has nontrivial singular values that reflect the geometry of the map relative to the chosen function class.

2.4 RKHS and cross-covariance operators

We briefly review the tools we need. Readers familiar with kernel methods may skim this section.

Kernels and feature maps

A *positive definite kernel* $k\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a function satisfying $\sum_{i,j} c_i c_j k(x_i, x_j) \geq 0$ for all finite collections $\{x_i\}$ and coefficients $\{c_i\}$. Every such kernel defines a unique RKHS $\mathcal{H}_k$ of functions $\mathbb{R}^d \to \mathbb{R}$ with inner product satisfying the *reproducing property*: $f(x) = \langle f, k(\cdot, x)\rangle_{\mathcal{H}_k}$.

The map $\Phi\colon x \mapsto k(\cdot, x)$ is the *feature map*; it embeds data points into $\mathcal{H}_k$. The inner product in feature space is the kernel evaluation: $\langle \Phi(x), \Phi(x')\rangle = k(x, x')$.

Common kernels include the Gaussian RBF $k(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$ and the Matérn family. The kernel is a free parameter: different kernels probe different aspects of the data.

Cross-covariance operators

Let $x \sim \mu$ be a random variable on X and $h = \psi(x)$ its image in H. The *cross-covariance operator* $C_{XH}\colon \mathcal{H}_H \to \mathcal{H}_X$ is defined by the bilinear form

$$\langle g, C_{XH} f\rangle_{\mathcal{H}_X} = \operatorname{Cov}\big(g(x),\ f(\psi(x))\big) = \mathbb{E}[g(x) f(\psi(x))] - \mathbb{E}[g(x)]\,\mathbb{E}[f(\psi(x))] \tag{1}$$

for all $g \in \mathcal{H}_X$, $f \in \mathcal{H}_H$. The operator exists and is bounded whenever the kernels are bounded (the bilinear form is continuous by the reproducing property and Cauchy-Schwarz).

Proposition 2.2 (Compactness). *If k_X and k_H are bounded kernels and μ has bounded support or $\mathbb{E}[k_X(x,x)] < \infty$ and $\mathbb{E}[k_H(h,h)] < \infty$ (which is automatic for bounded kernels), then C_{XH} is Hilbert-Schmidt (hence compact).*

Proof. The Hilbert-Schmidt norm satisfies $\|C_{XH}\|_{\text{HS}}^2 = \mathbb{E}_{x,x'}[k_X(x,x') k_H(\psi(x), \psi(x'))] - (\text{centering terms})$, which is finite when both kernels are bounded. Hilbert-Schmidt operators are compact. □

Since C_{XH} is compact, it admits a singular value decomposition (SVD).

Kernel CCA

Canonical correlation analysis (CCA) finds pairs of functions $(g, f) \in \mathcal{H}_X \times \mathcal{H}_H$ that maximize correlation (Bach & Jordan [2]; see also Bach [3] for connections to information theory):

$$\max_{g,f}\ \operatorname{Corr}\big(g(x),\ f(\psi(x))\big) \quad \text{subject to} \quad \operatorname{Var}(g(x)) = 1,\ \operatorname{Var}(f(\psi(x))) = 1. \tag{2}$$

Writing C_{XX} and C_{HH} for the auto-covariance operators on $\mathcal{H}_X$ and $\mathcal{H}_H$ respectively, the variance constraints become $\langle g, C_{XX} g\rangle = 1$ and $\langle f, C_{HH} f\rangle = 1$. The optimization is equivalent to the SVD of the *whitened cross-covariance operator*:

$$\widetilde{C}_{XH} = C_{XX}^{-1/2}\, C_{XH}\, C_{HH}^{-1/2} \colon \mathcal{H}_H \to \mathcal{H}_X. \tag{3}$$

The singular values $\sigma_1 \geq \sigma_2 \geq \cdots$ of $\widetilde{C}_{XH}$ are the *canonical correlations*, satisfying $0 \leq \sigma_j \leq 1$. The corresponding left and right singular vectors (after un-whitening) are the *canonical variates* in $\mathcal{H}_X$ and $\mathcal{H}_H$.

In practice, C_{XX} and C_{HH} are not invertible and we use Tikhonov regularization: replace $C_{XX}^{-1/2}$ with $(C_{XX} + \lambda I)^{-1/2}$ and similarly for C_{HH}.

2.5 Factored computations and composition

A *factored computation* is a pair of maps

$$X \xrightarrow{\psi} H \xrightarrow{\varphi} Y$$

with end-to-end map $f = \varphi \circ \psi$. We fix kernels k_X, k_H, k_Y on the three spaces. The whitened cross-covariance operators of the encoder and decoder are $\widetilde{C}_{XH}$ and $\widetilde{C}_{HY}$ respectively (defined as in (3)).

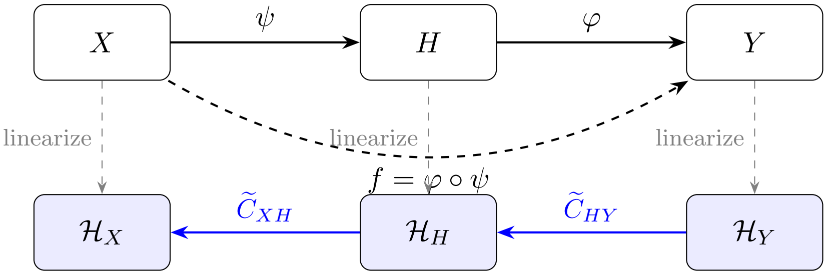


Figure 4: A factored computation $X \to H \to Y$ and its RKHS linearization. Each map is "linearized" by passing to the corresponding cross-covariance operator between reproducing kernel Hilbert spaces.

The composition theorem

The whitened cross-covariance operators compose correctly under the factorization assumption. We state the result in the empirical (finite-sample) setting, where all operators are finite-rank and inversion is well-defined.

Theorem 2.3 (Composition under Markov structure)**.** *Let $\{x_n\}_{n=1}^N$ be a finite sample and let all covariance operators be the corresponding empirical operators (which are finite-rank). If Y depends on X only through H (i.e., the Markov condition $X \perp Y \mid H$ holds), then*

$$\widetilde{C}_{XY} = \widetilde{C}_{XH}\, \widetilde{C}_{HY}. \tag{4}$$

That is, the whitened end-to-end operator equals the product of the whitened encoder and decoder operators.

Proof. For deterministic factored computations, $Y = \varphi(\psi(X))$, so Y is a deterministic function of $H = \psi(X)$. The conditional independence $X \perp Y \mid H$ follows: given H, the value of $Y = \varphi(H)$ is determined, so knowing X additionally provides no information about Y.

Under this Markov condition, the kernel Bayes rule (Fukumizu et al. [4]; Song et al. [11]) gives:

$$C_{XY} = C_{XH}\, C_{HH}^{-1}\, C_{HY}. \tag{5}$$

In the empirical setting, C_{HH} is a finite-rank operator on the N-dimensional subspace spanned by the kernel evaluations $\{k_H(\cdot, h_n)\}$, and its inverse is taken on this subspace.

Whitening both sides:

$$\begin{aligned} \widetilde{C}_{XY} &= C_{XX}^{-1/2}\, C_{XY}\, C_{YY}^{-1/2} && (6)\\ &= C_{XX}^{-1/2}\, C_{XH}\, C_{HH}^{-1}\, C_{HY}\, C_{YY}^{-1/2} && (7)\\ &= \big(C_{XX}^{-1/2}\, C_{XH}\, C_{HH}^{-1/2}\big)\big(C_{HH}^{-1/2}\, C_{HY}\, C_{YY}^{-1/2}\big) && (8)\\ &= \widetilde{C}_{XH}\, \widetilde{C}_{HY}. && \square \end{aligned}$$

Remark 2.4 (Population version)*.* In the population (infinite-dimensional) setting, C_{HH}^{-1} is unbounded for compact operators, so the composition (4) must be understood in the regularized limit: replace C_{HH}^{-1} with $(C_{HH} + \lambda I)^{-1}$ throughout, and the identity holds in the limit $\lambda \to 0^+$ on the range of C_{HH}. All empirical computations in this paper use Tikhonov regularization (§5.3), so this subtlety does not affect practical results.

Remark 2.5 (When is the Markov condition satisfied?)*.* For deterministic factored computations $Y = \varphi(\psi(X))$, the condition $X \perp Y \mid H$ always holds. This covers the main use case: splitting a neural network at any intermediate layer. The condition fails when Y depends on X through paths that bypass H (e.g., skip connections). In that case, $\widetilde{C}_{XH}\widetilde{C}_{HY}$ captures only the information routed through H, and the residual $\widetilde{C}_{XY} - \widetilde{C}_{XH}\widetilde{C}_{HY}$ quantifies the bypass contribution.

3 The Routing Matrix

3.1 Definition

Let the whitened encoder and decoder operators have SVDs

$$\widetilde{C}_{XH} = \sum_j \sigma_j\, u_j \otimes w_j, \qquad u_j \in \mathcal{H}_X,\ w_j \in \mathcal{H}_H, \tag{9}$$

$$\widetilde{C}_{HY} = \sum_k \tau_k\, \tilde{w}_k \otimes v_k, \qquad \tilde{w}_k \in \mathcal{H}_H,\ v_k \in \mathcal{H}_Y, \tag{10}$$

where $\{u_j\}$, $\{w_j\}$, $\{\tilde{w}_k\}$, $\{v_k\}$ are orthonormal systems in their respective RKHSs. Here $u \otimes w$ denotes the rank-one operator $f \mapsto \langle w, f\rangle_{\mathcal{H}_H}\, u$.

Definition 3.1 (Routing matrix)**.** The **routing matrix** of the factored computation $X \overset{\psi}{\to}$

$H \xrightarrow{\varphi} Y$ is the matrix

$$R_{jk} = \langle w_j,\, \tilde{w}_k \rangle_{\mathcal{H}_H}. \tag{11}$$

The routing matrix measures the overlap in $\mathcal{H}_H$ between the encoder's output modes and the decoder's input modes.

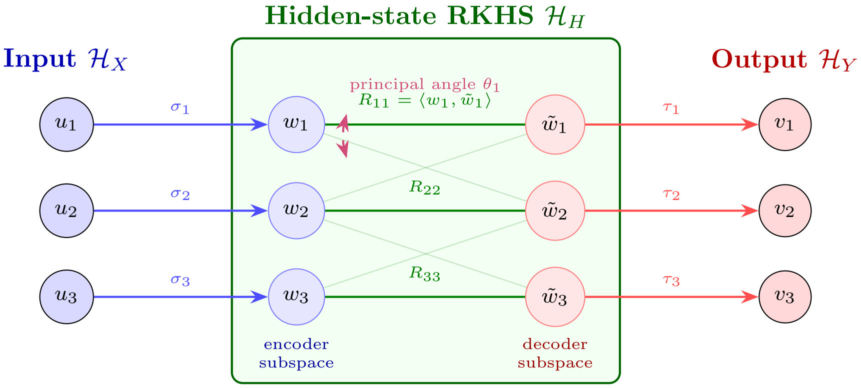


Figure 5: The routing matrix as subspace geometry in $\mathcal{H}_H$. Encoder modes $\{w_j\}$ and decoder modes $\{\tilde{w}_k\}$ live in the same hidden-state RKHS; $R_{jk} = \langle w_j, \tilde{w}_k \rangle$ measures their overlap, and principal angles θ_j quantify the alignment between encoder and decoder subspaces.

Both $\{w_j\}$ and $\{\tilde{w}_k\}$ live in the same space $\mathcal{H}_H$. The routing matrix $R_{jk} = \langle w_j, \tilde{w}_k \rangle_{\mathcal{H}_H}$ measures how well the encoder's j-th mode aligns with the decoder's k-th mode. Its singular values are the cosines of the *principal angles* between the encoder and decoder subspaces — a gauge-invariant geometric description of how the computation wires information.

3.2 Principal angles and the geometry of routing

The routing matrix $R_{jk} = \langle w_j, \tilde{w}_k \rangle_{\mathcal{H}_H}$ is the matrix of inner products between two orthonormal systems in the same Hilbert space. Its SVD

$$R = U_R \, \mathrm{diag}(\cos\theta_1, \ldots, \cos\theta_p) \, V_R^\top \tag{12}$$

recovers the *principal angles* $0 \le \theta_1 \le \cdots \le \theta_p \le \pi/2$ between the encoder subspace $\mathcal{W} = \mathrm{span}(\{w_j\}) \subset \mathcal{H}_H$ and the decoder subspace $\widetilde{\mathcal{W}} = \mathrm{span}(\{\tilde{w}_k\}) \subset \mathcal{H}_H$. This is the CS decomposition (Jordan, 1875; Björck & Golub [15]).

Proposition 3.2 (Principal angles of the routing matrix)**.** *Let* $\cos\theta_1 \ge \cdots \ge \cos\theta_p$ *be the singular values of* R*. Then:*

(a) $\cos\theta_i = 1$ *(*$\theta_i = 0$*) if and only if a one-dimensional subspace is shared between* $\mathcal{W}$ *and* $\widetilde{\mathcal{W}}$*: some encoder direction is exactly a decoder direction.*

(b) $\cos\theta_i = 0$ *(*$\theta_i = \pi/2$*) if and only if a dimension of encoder space is orthogonal to all of decoder space: information encoded along that direction is not used by the decoder.*

(c) The principal angles are gauge-invariant (they are singular values of the gauge-invariant matrix R*).*

(d) $\|R\|_F^2 = \sum_i \cos^2\theta_i$, *so the Frobenius norm of the routing matrix is a function of the principal angles alone.*

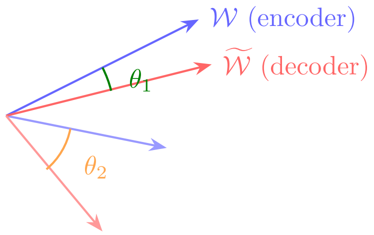
Proof. Parts (a) and (b) are standard properties of principal angles between subspaces. Part (c) follows from Proposition 3.6 (R is gauge-invariant) and the fact that singular values are continuous functions of matrix entries. Part (d) is $\|R\|_F^2 = \mathrm{tr}(R^\top R) = \sum_i \sigma_i(R)^2 = \sum_i \cos^2\theta_i$. □

The principal angles give a basis-independent description of how the encoder and decoder subspaces sit relative to each other in $\mathcal{H}_H$:

- All $\theta_i \approx 0$ (R has all singular values near 1): the encoder and decoder subspaces are nearly aligned. Every direction the encoder writes to is read by the decoder.
- Some $\theta_i \approx \pi/2$ (R has small singular values): part of the encoder subspace is nearly orthogonal to the decoder subspace. The encoder stores information that the decoder ignores.
- For comparing two systems: the principal angle spectra $\{\theta_i^{(1)}\}$ and $\{\theta_i^{(2)}\}$ (each computed from the SVD of the respective routing matrix) are more robust than entry-wise comparison of R^1 and R^2, because they depend only on the encoder-decoder subspace geometry, not on the ordering of modes within them.

Remark 3.3 (Principal angles between routing matrices). Principal angles can also be used to compare two routing matrices directly. Given $R^{(1)} \in \mathbb{R}^{p_1 \times q_1}$ and $R^{(2)} \in \mathbb{R}^{p_2 \times q_2}$ for two systems with aligned I/O modes ($P^X = P^Y = I$), the principal angles between the column spaces of $R^{(1)}$ and $R^{(2)}$ (viewed as subspaces of $\mathbb{R}^{\max(p_1,p_2)}$) measure how similar the two systems' routing structures are. Systems with aligned routing column spaces route information through similar subspaces of the encoder modes, regardless of the specific weights. This is more robust than $\|R^{(1)} - R^{(2)}\|_F$, which is sensitive to mode ordering and scale.

For two different observables Y_1, Y_2 on the same model (where the encoder modes are shared), the principal angles between the column spaces of $R^{(Y_1)}$ and $R^{(Y_2)}$ measure how much the two behaviors use overlapping subspaces of the encoder modes. Column spaces that are nearly aligned indicate shared computational structure; large principal angles indicate behavior-specific routing through different encoder subspaces.



θ_1 small: first encoder direction nearly aligned with decoder
θ_2 large: second encoder direction poorly read by decoder

Figure 6: Principal angles between encoder and decoder subspaces in $\mathcal{H}_H$. A small angle θ_1 means the first encoder direction is nearly aligned with the decoder; a large angle θ_2 means the second direction is poorly read by the decoder.

3.3 Decomposition of the end-to-end operator

Proposition 3.4 (Routing decomposition)**.** *The end-to-end whitened operator decomposes as*

$$\widetilde{C}_{XY} = \widetilde{C}_{XH}\,\widetilde{C}_{HY} = \sum_{j,k} \sigma_j\, R_{jk}\, \tau_k\, u_j \otimes v_k. \tag{13}$$

Proof. Expand the product using the SVDs:

$$\widetilde{C}_{XH}\,\widetilde{C}_{HY} = \Big(\sum_j \sigma_j\, u_j \otimes w_j\Big)\Big(\sum_k \tau_k\, \tilde{w}_k \otimes v_k\Big) \tag{14}$$

$$= \sum_{j,k} \sigma_j \tau_k\, (u_j \otimes w_j)(\tilde{w}_k \otimes v_k) \tag{15}$$

$$= \sum_{j,k} \sigma_j \tau_k\, \langle w_j, \tilde{w}_k\rangle_{\mathcal{H}_H}\, u_j \otimes v_k \tag{16}$$

$$= \sum_{j,k} \sigma_j\, R_{jk}\, \tau_k\, u_j \otimes v_k, \tag{17}$$

where we used the composition rule $(u \otimes w)(\tilde{w} \otimes v) = \langle w, \tilde{w}\rangle\, u \otimes v$ for rank-one operators. □

3.4 Gauge invariance

The hidden state H is defined only up to reparameterization. A diffeomorphism $\Gamma\colon H \to H'$ yields an equivalent computation $X \xrightarrow{\Gamma\circ\psi} H' \xrightarrow{\varphi\circ\Gamma^{-1}} Y$ with the same end-to-end behavior. The routing matrix should be invariant under such reparameterizations.

Definition 3.5 (Kernel-isometric reparameterization)**.** A diffeomorphism $\Gamma\colon H \to H$ is *kernel-isometric* if $k_H(\Gamma(h_1), \Gamma(h_2)) = k_H(h_1, h_2)$ for all h_1, h_2.

For translation-invariant kernels $k_H(h_1, h_2) = m(\|h_1 - h_2\|)$, the kernel-isometric reparameterizations are the rigid motions (rotations and translations). This is the gauge group for translation-invariant kernels: rotations and translations of the hidden representation should not affect the analysis.

Proposition 3.6 (Gauge invariance)**.** *The routing matrix is invariant under kernel-isometric reparameterizations.*

Proof. Let Γ be kernel-isometric. Define the unitary operator $U_\Gamma\colon \mathcal{H}_H \to \mathcal{H}_H$ by $U_\Gamma f = f \circ \Gamma^{-1}$. This is unitary because the kernel is preserved:

$$\langle U_\Gamma f, U_\Gamma g\rangle_{\mathcal{H}_H} = \langle f \circ \Gamma^{-1}, g \circ \Gamma^{-1}\rangle_{\mathcal{H}_H} = \langle f, g\rangle_{\mathcal{H}_H}$$

(the last equality holds because Γ preserves the kernel, hence the RKHS inner product).

Under reparameterization by Γ:

- The cross-covariance transforms as $C_{XH'} = C_{XH}\, U_\Gamma^*$ (the feature map on H' is $\Phi_{H'}(\Gamma(h)) = k_H(\cdot, \Gamma(h)) = k_H(\Gamma^{-1}(\cdot), h) = U_\Gamma k_H(\cdot, h) = U_\Gamma \Phi_H(h)$, so the cross-covariance picks up U_Γ on the H-side).

- The auto-covariance transforms as $C_{H'H'} = U_\Gamma\, C_{HH}\, U_\Gamma^*$.
- The whitened operator becomes $\widetilde{C}_{XH'} = \widetilde{C}_{XH}\, U_\Gamma^*$.

The SVD of $\widetilde{C}_{XH'} = \widetilde{C}_{XH} U_\Gamma^*$ has right singular vectors $U_\Gamma w_j$ (since $\widetilde{C}_{XH} U_\Gamma^* (U_\Gamma w_j) = \widetilde{C}_{XH} w_j = \sigma_j u_j$).

Similarly, the whitened decoder operator picks up U_Γ on the left, giving left singular vectors $U_\Gamma \tilde{w}_k$.

Therefore:

$$R'_{jk} = \langle U_\Gamma w_j,\, U_\Gamma \tilde{w}_k \rangle_{\mathcal{H}_H} = \langle w_j,\, \tilde{w}_k \rangle_{\mathcal{H}_H} = R_{jk}$$

since U_Γ is unitary. □

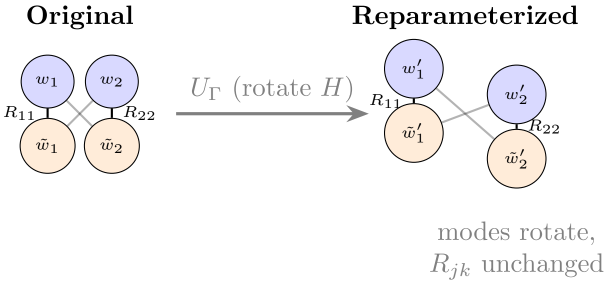


Figure 7: Gauge invariance of the routing matrix. A unitary reparameterization U_Γ of the hidden state rotates encoder and decoder modes jointly, but the inner products R_{jk} are preserved.

Theorem 3.7 (Complete gauge invariants)**.** *The tuple $(\sigma_j, \tau_k, R_{jk}, u_j, v_k)$ is the complete set of data invariant under RKHS-unitary reparameterizations of the hidden state. Precisely:*

(a) ***Invariance.*** *Under any RKHS unitary $U\colon \mathcal{H}_H \to \mathcal{H}_H$, the quantities σ_j, τ_k, R_{jk}, $u_j \in \mathcal{H}_X$, and $v_k \in \mathcal{H}_Y$ are unchanged.*

(b) ***Completeness.*** *If two factored computations $(\widetilde{C}_{XH}, \widetilde{C}_{HY})$ and $(\widetilde{C}'_{XH}, \widetilde{C}'_{HY})$ have the same tuple, then there exists a RKHS unitary U such that $\widetilde{C}'_{XH} = \widetilde{C}_{XH} U^*$ and $\widetilde{C}'_{HY} = U\, \widetilde{C}_{HY}$. (For the linear kernel $k_H(h, h') = h^\top h'$, every RKHS unitary is an orthogonal matrix, hence a kernel isometry, so this characterizes kernel-isometric equivalence.)*

Proof. **(a)** The singular values σ_j, τ_k and the input/output modes $u_j \in \mathcal{H}_X$, $v_k \in \mathcal{H}_Y$ live outside the hidden-state RKHS and are unaffected by U. For R_{jk}: under U, the encoder right singular vectors become $U w_j$ and the decoder left singular vectors become $U \tilde{w}_k$, so $R'_{jk} = \langle U w_j, U \tilde{w}_k \rangle_{\mathcal{H}_H} = \langle w_j, \tilde{w}_k \rangle_{\mathcal{H}_H} = R_{jk}$ by unitarity. This extends Proposition 3.6 from kernel isometries to all RKHS unitaries.

(b) Let $(\sigma_j, \tau_k, R_{jk}, u_j, v_k)$ be the shared tuple. The two computations have encoder right singular vectors $\{w_j\}$ and $\{w'_j\}$, and decoder left singular vectors $\{\tilde{w}_k\}$ and $\{\tilde{w}'_k\}$, all orthonormal in $\mathcal{H}_H$. Their pairwise inner products agree:

$$\langle w_j, \tilde{w}_k \rangle = R_{jk} = \langle w'_j, \tilde{w}'_k \rangle, \qquad \langle w_j, w_{j'} \rangle = \delta_{jj'} = \langle w'_j, w'_{j'} \rangle,$$

and similarly for $\tilde{w}$–$\tilde{w}'$. Therefore the linear map $V: w_j \mapsto w'_j,\ \tilde{w}_k \mapsto \tilde{w}'_k$ is a partial isometry from $\operatorname{span}(\{w_j\} \cup \{\tilde{w}_k\})$ to $\operatorname{span}(\{w'_j\} \cup \{\tilde{w}'_k\})$. Since the domain and range subspaces have the same dimension (their Gram matrices are identical, hence have the same rank), any partial isometry between them extends to a unitary on the whole space (extending arbitrarily on the orthogonal complement, which has equal co-dimension in both cases — or is infinite-dimensional on both sides when $\mathcal{H}_H$ is infinite-dimensional). Thus there exists a RKHS unitary U with $Uw_j = w'_j$ and $U\tilde{w}_k = \tilde{w}'_k$. Then $\widetilde{C}'_{XH} = \sum_j \sigma_j\, u_j \otimes w'_j = \sum_j \sigma_j\, u_j \otimes U w_j = \widetilde{C}_{XH}\, U^*$, and similarly $\widetilde{C}'_{HY} = U\, \widetilde{C}_{HY}$.

(c) For the linear kernel, $\mathcal{H}_H = \mathbb{R}^{d_H}$ with the Euclidean inner product. Every unitary on a finite-dimensional real inner product space is an orthogonal matrix, and every orthogonal matrix O defines a kernel isometry via $\Gamma(h) = Oh$. □

Remark 3.8 (Relation to unitary invariants). For bounded linear operators between Hilbert spaces, two operators A and B satisfy $A = UBV^*$ for unitaries U, V if and only if $\sigma_j(A) = \sigma_j(B)$ for all j. The singular value sequence is the complete invariant under bilateral unitary equivalence (von Neumann [22]).

Theorem 3.7 is the factored analogue: for factored computations, the tuple (σ, τ, R, u, v) is the complete invariant under hidden-state reparameterization. Any gauge-invariant quantity — the behavioral distance of §4.3, the Eckart-Young error of §5 — must be a function of this tuple. The canonical correlations σ_j, τ_k play the role of singular values; the routing matrix R_{jk} captures the additional structure present in factored operators (where the hidden state introduces a degree of freedom absent from unfactored ones).

3.5 Kernel monotonicity: a diagnostic for nonlinear structure

The kernel choice on the hidden state determines which features are visible. While routing matrices computed under different kernels cannot be compared entry-wise (they are expressed in different mode bases), the *spectral profiles* are comparable: the canonical correlations σ_j and τ_k are defined by an optimization problem (maximize correlation over the RKHS) whose values are well-defined scalars for each kernel choice.

Proposition 3.9 (Kernel monotonicity). *For any RKHS inclusion $\mathcal{H}_1 \subset \mathcal{H}_2$ (e.g., linear $\subset$ RBF), the canonical correlations satisfy $\sigma_j^{(1)} \leq \sigma_j^{(2)}$ for all j.*

Proof. The j-th canonical correlation is defined by maximizing correlation over functions orthogonal to the first $j-1$ modes. The optimization over the larger class $\mathcal{H}_2 \supset \mathcal{H}_1$ can only achieve a higher maximum. □

This yields a *spectral diagnostic for the linear representation hypothesis*: compute encoder canonical correlations with a linear kernel and an RBF kernel on H. If $\sigma_j^{\text{lin}} \approx \sigma_j^{\text{rbf}}$ for all j, the X-H dependence is well-captured by linear features. If the nonlinear kernel yields strictly higher correlations (especially beyond the first few modes), there is functionally relevant nonlinear structure invisible to linear probes. The gap $\sigma_j^{\text{rbf}} - \sigma_j^{\text{lin}}$ measures the nonlinear information at mode j.

3.6 Comparison to existing methods

CKA and SVCCA. CKA compares two representations by computing a scalar similarity score; the routing matrix provides a finer decomposition into per-mode contributions tied to the computation's I/O structure. Two hidden states can have high CKA but very different routing matrices. SVCCA [10] and CCA-based methods [8] decompose directional information via canonical correlations; our contribution beyond these is the routing matrix, which captures *wiring* between encoder and decoder modes.

Probing classifiers. A linear probe tests whether a feature is linearly decodable. The routing matrix addresses a prior question: which modes are actually *used* by the downstream computation? A feature may be decodable but carry negligible routing weight — the model encodes it but does not use it. The routing matrix distinguishes "present" from "functionally relevant."

Activation patching. Patching measures the causal effect of ablating a component and is cheap per hypothesis. The routing matrix is more expensive ($N \gg k^2/\text{gap}^2$ samples, $O(N^3)$ or $O(Np^2)$ computation) but provides a *global decomposition* of information flow rather than a per-component causal test. The CCA modes are distributed directions in RKHS, not individual neurons, so the two methods operate at different granularities.

3.7 Interpreting the CCA modes

The CCA directions $u_j, w_j, \tilde{w}_k, v_k$ are the modes of the decomposition, but what are they concretely?

Each CCA direction is an RKHS function. Concretely, $u_j \in \mathcal{H}_X$ is a function $u_j \colon X \to \mathbb{R}$ that assigns a scalar "activation" to each input. In the empirical setting, $u_j(x) = \sum_n \alpha_n^{(j)} k_X(x, x_n)$ — a kernel-weighted combination of similarities to training points. Its activation pattern across a dataset defines what "encoder mode j" means: inputs where u_j is large are the inputs that this mode responds to.

Linear kernel. For the linear kernel $k(x, x') = x^\top x'$, the RKHS is $\mathbb{R}^{d_X}$ with the Euclidean inner product, and the CCA directions reduce to linear directions in data space. The modes are the same objects that ordinary CCA or linear probes would find.

Nonlinear kernels. For nonlinear kernels (RBF, Matérn), the CCA directions are genuinely nonlinear functions of the data. The modes $w_j \in \mathcal{H}_H$ can capture features of the hidden state that are *not* linearly decodable — a hidden-state feature represented as a nonlinear manifold, a conjunction, or a curved surface in activation space will appear as a CCA mode provided it carries information detectable by the chosen kernel. (The kernel determines which nonlinear features are visible: an RBF kernel with mismatched bandwidth may miss structure at the wrong scale.)

Relation to SAE features. SAE features are directions in activation space — linear functionals on H. Each SAE feature d_i defines a function $h \mapsto d_i^\top h$ on the hidden state. For a *linear* kernel on H, where $\mathcal{H}_H = \mathbb{R}^{d_H}$, one can project SAE features onto the CCA basis by computing $\langle d_i^\top(\cdot), w_j \rangle_{\mathcal{H}_H} = d_i^\top H^\top \alpha^{(j)}$, where H is the $N \times d_H$ data matrix and $\alpha^{(j)}$ is the CCA coefficient vector. This gives each SAE feature a "routing profile" — its weight in each CCA mode — without additional experiments. Note: for a linear kernel on H, the CCA modes *are* linear directions, and this projection is a change of basis. The content beyond SAEs is specifically in the nonlinear kernel case, where the CCA modes are no longer linear functionals and the projection requires evaluating RKHS inner products rather than dot products.

4 The Routing Calculus

An important degree of freedom in the routing decomposition: the "output" Y in the factored computation $X \to H \to Y$ need not be the raw model output. It can be *any function of the output* that captures a behavior of interest. This section develops the theory of observable substitution, showing that different behavioral targets yield different routing matrices from the same encoder, and that these routing matrices compose algebraically under post-processing.

4.1 Definition and the encoder invariance principle

Formally, if $\eta\colon Y_{\text{raw}} \to Y_{\text{score}}$ is a behavioral scoring function, the modified computation is $X \xrightarrow{\psi} H \xrightarrow{\eta\circ\varphi} Y_{\text{score}}$, and all the theory applies with Y_{score} in place of Y. The Markov condition $X \perp Y_{\text{score}} \mid H$ still holds (since $Y_{\text{score}} = \eta(\varphi(H))$ is a function of H).

Theorem 4.1 (Encoder invariance)**.** *The encoder CCA between X and H depends only on the input distribution and the hidden state — not on which output observable is chosen. The encoder modes $\{w_j\}$ and correlations $\{\sigma_j\}$ are reused across all behavioral questions. Each new behavioral observable requires only a decoder CCA.*

Proof. The encoder CCA computes the SVD of $\widetilde{C}_{XH} = C_{XX}^{-1/2}\, C_{XH}\, C_{HH}^{-1/2}$, which depends only on the kernels k_X, k_H and the joint distribution of (X, H). The output Y does not appear in any of these quantities. □

The computational significance: the encoder step is $O(N^3)$ for exact kernel CCA (or $O(Np^2)$ with random Fourier features) and involves the high-dimensional hidden state. Each additional behavioral target requires only a decoder CCA on the output space.

Example 4.2 (Behavioral routing (illustrative, not yet tested))**.** The following is a *hypothetical* application to a language model, included to show what the framework would compute in principle. It has not been validated on transformer language models (see Limitations, §6). In transformers, skip connections violate the Markov assumption at every layer, so the routing matrix would capture only the incremental contribution of each layer's nonlinearity, not the full information flow.

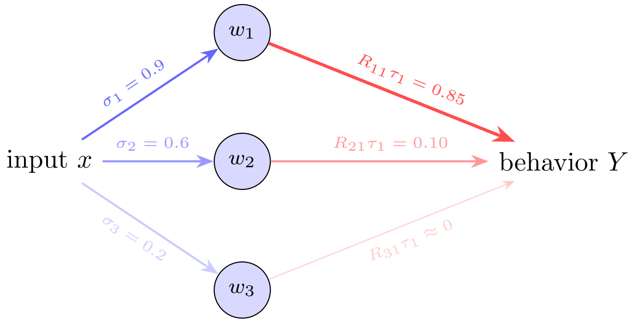
With that caveat, in principle we could set:

- Y = next-token logits ⇒ routing matrix for prediction.
- Y = logit of a specific token ⇒ routing for a particular prediction.
- Y = a linear probe's output for truthfulness ⇒ routing for truthful behavior.
- Y = a toxicity score ⇒ routing for toxic content.

Each choice would give a *different routing matrix.* Whether kernel CCA modes correspond to semantically meaningful features in this setting is an open empirical question.

4.2 Observable composition

Observable substitution composes: post-processing an observable yields a routing matrix that factors through the original.



Each arrow is a gauge-invariant quantity with estimation error bounds (Corollary 5.8).

"Mode w_1 carries $\sigma_1 = 0.9$ of the input structure and routes $R_{11}\tau_1 = 0.85$ of it to behavior Y."

Figure 8: Behavior-specific attribution via routing. Each arrow carries a gauge-invariant quantity with estimation error bounds: σ_j measures how much input structure encoder mode j captures, and $R_{jk}\tau_k$ measures how much of that reaches behavior Y.

Theorem 4.3 (Observable composition)**.** *Let $X \to H \to Y$ satisfy $X \perp Y \mid H$, and let $L\colon Y \to Z$ be a measurable post-processing map. Then*

$$\widetilde{C}_{X,L(Y)} = \widetilde{C}_{XY} \cdot \widetilde{C}_{Y,L(Y)}. \tag{18}$$

Proof. We verify three Markov conditions and apply the composition theorem (Theorem 2.3) twice.

Step 1. $X \perp L(Y) \mid H$. Since $Y = \varphi(H)$, we have $L(Y) = L(\varphi(H))$, which is a deterministic function of H. Hence $X \perp L(Y) \mid H$ follows from $X \perp Y \mid H$ and the data-processing inequality for conditional independence.

Step 2. $H \perp L(Y) \mid Y$. Since $L(Y)$ is $\sigma(Y)$-measurable, conditioning on Y makes $L(Y)$ a constant, so $H \perp L(Y) \mid Y$ holds trivially.

Step 3. Applying the composition theorem to the chain $X \to H \to L(Y)$:

$$\widetilde{C}_{X,L(Y)} = \widetilde{C}_{XH} \cdot \widetilde{C}_{H,L(Y)}.$$

Applying it again to the chain $H \to Y \to L(Y)$:

$$\widetilde{C}_{H,L(Y)} = \widetilde{C}_{HY} \cdot \widetilde{C}_{Y,L(Y)}.$$

Substituting and using $\widetilde{C}_{XH} \cdot \widetilde{C}_{HY} = \widetilde{C}_{XY}$ (the original composition theorem):

$$\widetilde{C}_{X,L(Y)} = \widetilde{C}_{XY} \cdot \widetilde{C}_{Y,L(Y)}.$$

□

Corollary 4.4 (DPI for observables)**.** *Post-processing the observable can only lose information* *about the input:*

$$\sigma_j(\widetilde{C}_{X,L(Y)}) \leq \sigma_j(\widetilde{C}_{XY}) \cdot \|\widetilde{C}_{Y,L(Y)}\|_{\mathrm{op}}.$$

Proof. Apply the Ky Fan singular value inequality (Bhatia [26], Theorem IV.2.5): $\sigma_j(AB) \leq \sigma_j(A)\|B\|_{\mathrm{op}}$ to (18). □

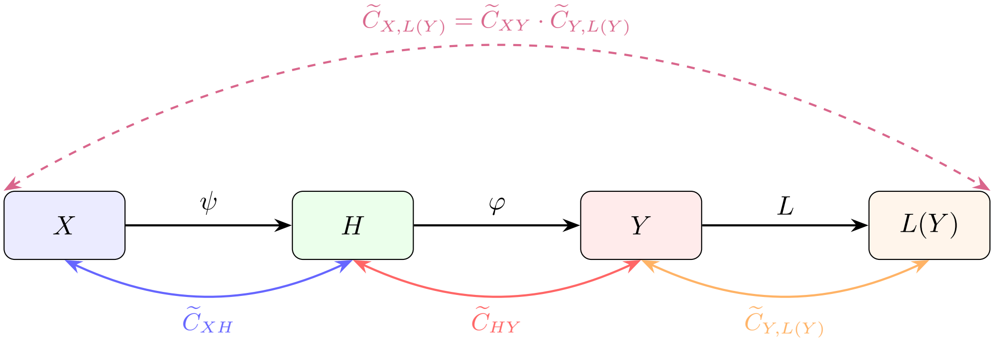


Figure 9: Observable composition chain. Post-processing Y through a map L yields a composed whitened cross-covariance $\widetilde{C}_{X,L(Y)} = \widetilde{C}_{XY} \cdot \widetilde{C}_{Y,L(Y)}$, avoiding encoder recomputation.

4.3 Comparing hidden states

The previous subsections decompose a *single* computation into modes. Now we address comparison: given two systems that perform the same (or similar) task, how do we compare their internal organization? The comparison decomposes into three levels — spectral, alignment, and routing — that progressively refine the question from "do they carry the same amount of information?" to "do they wire it the same way?"

Consider two systems with common I/O spaces:

$$\text{System } i\colon \quad X \xrightarrow{\psi_i} H_i \xrightarrow{\varphi_i} Y, \qquad i = 1, 2.$$

The hidden spaces $H_i = \mathbb{R}^{d_{H_i}}$ may have different dimensions. Each system has its own triple $(\boldsymbol{\sigma}^i, R^i, \boldsymbol{\tau}^i)$ and I/O mode systems $\{u^i_j\} \subset \mathcal{H}_X$, $\{v^i_k\} \subset \mathcal{H}_Y$.

The three-level comparison

Definition 4.5 (I/O overlap matrices)**.** The **input overlap matrix** and **output overlap matrix** are

$$P^X_{jj'} = \langle u^1_j, u^2_{j'} \rangle_{\mathcal{H}_X}, \qquad P^Y_{kk'} = \langle v^1_k, v^2_{k'} \rangle_{\mathcal{H}_Y}.$$

The comparison of two computations decomposes into three levels:

1. **Spectral comparison:** Are $\boldsymbol{\sigma}^1 \approx \boldsymbol{\sigma}^2$ and $\boldsymbol{\tau}^1 \approx \boldsymbol{\tau}^2$? This asks whether the two systems carry the same amount of information per mode – the "bandwidth" of the computation.
2. **Mode alignment:** Are $P^X \approx I$ and $P^Y \approx I$? This asks whether the two systems use the same features at the I/O level.
3. **Routing comparison:** After aligning modes, is $R^1 \approx (P^X)^\top R^2 P^Y$? This asks whether the two systems wire information the same way through their hidden states.

Behavioral distance

Definition 4.6 (Behavioral distance)**.** The **behavioral distance** is

$$d_{\mathrm{IO}} = \|\widetilde{C}^1_{XY} - \widetilde{C}^2_{XY}\|_{\mathrm{HS}},$$

the Hilbert-Schmidt norm distance between the end-to-end operators.

This is zero if and only if the systems have identical I/O behavior (in the kernel CCA sense). It is a pseudometric, inheriting symmetry and the triangle inequality from the Hilbert-Schmidt norm.

Interpretation. The HS norm $\|\widetilde{C}^1_{XY} - \widetilde{C}^2_{XY}\|_{\mathrm{HS}}$ measures the total disagreement between the two systems' whitened input-output maps, summed over all mode pairs. It decomposes into spectral and routing contributions (Proposition 4.7).

Decomposition of behavioral distance

Proposition 4.7 (Behavioral distance decomposition)**.**

$$d^2_{\mathrm{IO}} = \sum_{j,k} (\sigma^1_j R^1_{jk} \tau^1_k)^2 + \sum_{j',k'} (\sigma^2_{j'} R^2_{j'k'} \tau^2_{k'})^2 - 2 \sum_{\substack{j,j' \\ k,k'}} \sigma^1_j R^1_{jk} \tau^1_k \cdot \sigma^2_{j'} R^2_{j'k'} \tau^2_{k'} \cdot P^X_{jj'} \, P^Y_{kk'}. \tag{19}$$

Proof. Write $\widetilde{C}^i_{XY} = \sum_{j,k} \sigma^i_j R^i_{jk} \tau^i_k \, u^i_j \otimes v^i_k$ (Proposition 3.4). The HS norm squared is:

$$\|\widetilde{C}^1_{XY} - \widetilde{C}^2_{XY}\|^2_{\mathrm{HS}} = \mathrm{tr}\big((\widetilde{C}^1_{XY})^* \widetilde{C}^1_{XY}\big) + \mathrm{tr}\big((\widetilde{C}^2_{XY})^* \widetilde{C}^2_{XY}\big) - 2 \,\mathrm{Re}\,\mathrm{tr}\big((\widetilde{C}^1_{XY})^* \widetilde{C}^2_{XY}\big). \tag{20}$$

The self-terms give $\sum_{j,k} (\sigma^i_j R^i_{jk} \tau^i_k)^2$ (since $\{u^i_j \otimes v^i_k\}$ are orthonormal in HS norm, being tensor products of orthonormal systems).

The cross-term:

$$\mathrm{tr}\big((\widetilde{C}^1)^* \widetilde{C}^2\big) = \sum_{\substack{j,j' \\ k,k'}} \sigma^1_j R^1_{jk} \tau^1_k \cdot \sigma^2_{j'} R^2_{j'k'} \tau^2_{k'} \cdot \underbrace{\langle u^1_j, u^2_{j'} \rangle}_{P^X_{jj'}} \underbrace{\langle v^1_k, v^2_{k'} \rangle}_{P^Y_{kk'}}. \qquad \square$$

When the I/O modes are aligned ($P^X = P^Y = I$), the cross-terms collapse and the distance becomes a sum of squared differences $d^2_{\mathrm{IO}} = \sum_{j,k} (\sigma^1_j R^1_{jk} \tau^1_k - \sigma^2_j R^2_{jk} \tau^2_k)^2$, cleanly separating spectral and routing contributions.

4.4 Cross-system translation

The behavioral distance d_{IO} tells us *how different* two computations are. A finer tool: decompose the information-theoretic cost of translating one system's hidden state into the other's, separating the cost into an *encoder gap* (information irreversibly discarded by the encoder) and a *translation gap* (information lost in cross-system mapping).

Why canonical correlations measure information. The j-th canonical correlation σ_j between X and H is defined as the maximum correlation achievable by RKHS functions $f \in \mathcal{H}_X$, $g \in \mathcal{H}_H$ that are orthogonal to the first $j-1$ optimal pairs: $\sigma_j = \max_{f \perp u_1,\ldots,u_{j-1}} \max_{g \perp w_1,\ldots,w_{j-1}} \operatorname{Corr}(f(X), g(H))$. For jointly Gaussian variables with a finite-dimensional kernel, each canonical correlation captures one "bit" of mutual information: $I(X;H) = -\frac{1}{2}\sum_j \log(1-\sigma_j^2)$. In general, σ_j measures the strength of the j-th independent channel of statistical dependence between the two spaces. The DPI then says that each of these channels can only lose strength as information passes through a computation — the "information ellipsoid" can only shrink.

Consider two systems on common inputs:

$$\text{System 1:} \quad X \xrightarrow{\psi} H \xrightarrow{\varphi} Y, \qquad \text{System 2:} \quad X \xrightarrow{\psi'} H' \xrightarrow{\varphi'} Y'.$$

The hidden spaces H, H' may have different dimensions. The outputs Y, Y' may differ (different tasks) or coincide (same task, different architectures).

We ask: how well can System 1's hidden state $H = \psi(X)$ support System 2's output Y'?

The DPI chain and information decomposition

For any translation map $T\colon H \to H'$, the chain $X \to H \xrightarrow{T} T(H)$ is a Markov chain. The data processing inequality applied to the canonical correlations with Y' gives:

$$\sigma_j(X;\,Y') \;\geq\; \sigma_j(H;\,Y') \;\geq\; \sigma_j(T(H);\,Y'). \tag{21}$$

Here $\sigma_j(A;B)$ denotes the j-th canonical correlation between A and B (the j-th singular value of the whitened cross-covariance operator between the two). These are computable via kernel CCA on shared samples.

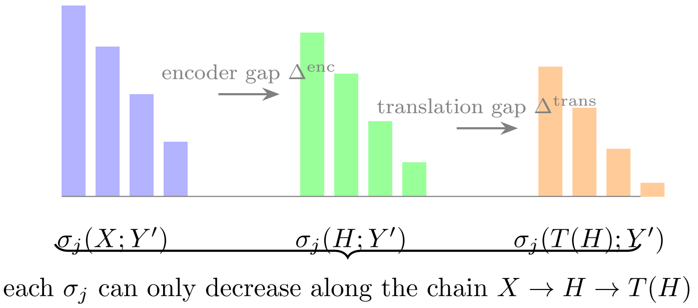


Figure 10: Spectral data processing inequality. Canonical correlations σ_j can only decrease along the chain $X \to H \to T(H)$. The encoder gap measures information lost by the encoder; the translation gap measures additional loss from the translation map.

The two gaps in (21) have distinct interpretations:

Definition 4.8 (Information decomposition)**.** The **encoder gap** at mode j is

$$\Delta_j^{\text{enc}} = \sigma_j(X;\,Y') - \sigma_j(H;\,Y').$$

This is the Y'-relevant information that X carries but that System 1's encoder ψ irreversibly discarded. It depends only on ψ and Y', not on the translation T.

The **translation gap** at mode j is

$$\Delta_j^{\text{trans}} = \sigma_j(H;\, Y') - \sigma_j(T(H);\, Y').$$

This is the additional Y'-relevant information lost by the translation. The optimal T^* minimizes this.

The total deficit $\sigma_j(X;Y') - \sigma_j(T(H);Y') = \Delta_j^{\text{enc}} + \Delta_j^{\text{trans}}$ is the price of using System 1's translated state instead of raw input to predict System 2's output.

The middle term: cross-system predictive content

The quantity $\sigma_j(H;Y')$ — the canonical correlations between System 1's hidden state and System 2's output — is the key diagnostic. It measures how much Y'-relevant information System 1's hidden state *happens to carry*, even though System 1 was not trained for Y'.

This is computable: run kernel CCA between $H = \psi(X)$ and $Y' = \varphi'(\psi'(X))$ on shared input samples. No translation map is needed.

- $\sigma_j(H;Y') \approx \sigma_j(X;Y')$ for all j: System 1's encoder retains nearly all Y'-relevant information. The encoder gap is small. Good translation to Y' is possible in principle.
- $\sigma_j(H;Y') \ll \sigma_j(X;Y')$: System 1's encoder has irreversibly discarded Y'-relevant information. No translation can recover it. This is the fundamental limit on cross-system transfer.
- When $Y = Y'$ (same task): the encoder gap measures how much each architecture's inductive bias costs in terms of the other's preferred features. This is a per-mode measure of architectural compatibility.

Optimal translation

The optimal translation $T^*\colon \mathcal{H}_H \to \mathcal{H}_{H'}$ maximizes the Y'-predictive content of the translated state. It operates between different RKHSs (different dimensions, different kernels) and reduces to a matrix problem on Gram matrices in the empirical setting.

Proposition 4.9 (Translation bound)**.** *For the optimal translation T^*, the translated canonical correlations satisfy*

$$\sigma_j(T^*(H);\, Y') \leq \sigma_j(H;\, Y'),$$

with equality when T^* is lossless on the Y'-relevant subspace of $\mathcal{H}_H$. In particular, the translation gap vanishes when the Y'-relevant modes of H can be perfectly mapped to Y'-relevant modes of H'.

Proof. The chain $X \to H \to T^*(H)$ is Markov, so the DPI gives the inequality. For equality: if T^* restricted to the span of the top Y'-relevant modes of H is an isometry into $\mathcal{H}_{H'}$ (preserving inner products on this subspace), then the whitened cross-covariance $\widetilde{C}_{T^*(H)Y'}$ has the same singular values as $\widetilde{C}_{HY'}$ on those modes. □

Routing structure of translation

The routing matrix of the *stitched computation* $X \to H \xrightarrow{T^*} T^*(H) \to Y'$ reveals which of System 1's encoder modes are useful for System 2's task:

- Compute kernel CCA between X and H: encoder modes $\{w_j\}$ with correlations $\{\sigma_j\}$.
- Compute kernel CCA between $T^*(H)$ and Y': translated decoder modes $\{\tilde{w}'_k\}$ with correlations $\{\tau'_k\}$.
- The *translation routing matrix* $R^{1\to 2}_{jk} = \langle w_j, (T^*)^* \tilde{w}'_k \rangle_{\mathcal{H}_H}$ captures how System 1's encoder modes connect to System 2's decoder modes through the translation.

Comparing $R^{1\to 2}$ with System 2's native routing matrix R' identifies which routing pathways survive translation (shared computational structure) and which are disrupted (architecture-specific wiring).

5 Guarantees

5.1 Optimality: Eckart-Young for factored operators

The Eckart-Young theorem guarantees that truncating to the top k singular values gives the best rank-k approximation. We extend this to the factored setting, showing that the routing matrix determines which modes actually matter.

For a compact operator $A = \sum_i \sigma_i \, u_i \otimes v_i$ between Hilbert spaces, the best rank-k approximation in Hilbert-Schmidt norm is $A_k = \sum_{i=1}^k \sigma_i \, u_i \otimes v_i$, with $\|A - A_k\|^2_{\mathrm{HS}} = \sum_{i>k} \sigma_i^2$ and $\|A - A_k\|_{\mathrm{op}} = \sigma_{k+1}$ (Eckart & Young [20]; Mirsky [21]). In the factored setting:

Theorem 5.1 (Factored truncation bounds)**.** *Let $(\widetilde{C}_{XH})_{k_1}$ and $(\widetilde{C}_{HY})_{k_2}$ be the rank-k_1 and rank-k_2 truncations of the encoder and decoder operators, and let $\widehat{T} = (\widetilde{C}_{XH})_{k_1} \, (\widetilde{C}_{HY})_{k_2}$. Then:*

(a) Operator norm bound:

$$\|\widetilde{C}_{XY} - \widehat{T}\|_{\mathrm{op}} \le \sigma_{k_1+1} \, \tau_1 + \sigma_1 \, \tau_{k_2+1}. \tag{22}$$

(b) Hilbert-Schmidt bound:

$$\|\widetilde{C}_{XY} - \widehat{T}\|^2_{\mathrm{HS}} = \sum_{\substack{(j,k):\\ j>k_1 \text{ or } k>k_2}} (\sigma_j \, R_{jk} \, \tau_k)^2. \tag{23}$$

Proof. See Appendix A. □

Remark 5.2 (Routing-aware truncation)*.* The key insight of (23): discarding input mode j costs $\sum_k (\sigma_j R_{jk} \tau_k)^2$, which depends on how strongly mode j *routes to retained output modes.* A mode with large σ_j but $R_{jk} \approx 0$ for all retained k contributes little to the end-to-end computation and is cheap to discard.

This means that optimal truncation is a *joint* selection problem: which input and output modes to retain, weighted by the routing matrix. The naive strategy "keep the top k of each" is suboptimal when R has significant off-diagonal structure.

Note that the factored truncation $\widehat{T}$ is a *structured* truncation that respects the factored form $\widetilde{C}_{XH} \, \widetilde{C}_{HY}$. It is generally suboptimal compared to the unconstrained best rank-k approximation of $\widetilde{C}_{XY}$ (which is given by the standard Eckart-Young theorem applied directly to the end-to-end operator). The point is that the factored truncation preserves the factored structure: we want to understand which encoder and decoder modes matter, not just which end-to-end modes matter.

5.2 Temporal slicing and the spectral data processing inequality

When the two "systems" being compared are two snapshots of the same computation at different times (e.g., different layers of the same network), additional structure emerges.

At time t_1: $X \xrightarrow{\psi_1} H_1 \xrightarrow{\varphi_1} Y$. At time $t_2 > t_1$: $X \xrightarrow{\psi_2} H_2 \xrightarrow{\varphi_2} Y$. Each has its own triple $(\boldsymbol{\sigma}^i, R^i, \boldsymbol{\tau}^i)$.

The non-independence is expressed by a transition map $\Phi\colon H_1 \to H_2$ satisfying $\psi_2 = \Phi \circ \psi_1$ and $\varphi_1 = \varphi_2 \circ \Phi$.

The spectral data processing inequality

Theorem 5.3 (Spectral DPI for unwhitened cross-covariance)**.** *Let $K_\Phi\colon \mathcal{H}_H \to \mathcal{H}_H$ be the RKHS composition operator induced by Φ (via $K_\Phi g = g \circ \Phi$). Then the singular values of the* unwhitened *cross-covariance operators satisfy*

$$\sigma_j(C^2_{XH}) \leq \|K_\Phi\|_{\mathrm{op}}\, \sigma_j(C^1_{XH}) \quad \textit{for all } j, \tag{24}$$

$$\sigma_k(C^2_{HY}) \leq \|K_\Phi\|_{\mathrm{op}}\, \sigma_k(C^1_{HY}) \quad \textit{for all } k. \tag{25}$$

In particular, if K_Φ is a contraction ($\|K_\Phi\|_{\mathrm{op}} \leq 1$), both the encoder-side and decoder-side unwhitened singular values decrease with depth.

Proof. Encoder side. The unwhitened cross-covariance operators satisfy $C^2_{XH} = C^1_{XH}\, K_\Phi$, since $\psi_2 = \Phi \circ \psi_1$ implies $f(\psi_2(x)) = (K_\Phi f)(\psi_1(x))$ for all $f \in \mathcal{H}_H$. By the Ky Fan inequality $\sigma_j(AB) \leq \sigma_j(A)\, \|B\|_{\mathrm{op}}$:

$$\sigma_j(C^2_{XH}) = \sigma_j(C^1_{XH}\, K_\Phi) \leq \sigma_j(C^1_{XH})\, \|K_\Phi\|_{\mathrm{op}}.$$

Decoder side. For $f \in \mathcal{H}_H$ and $g \in \mathcal{H}_Y$: $\langle f, C^2_{HY}\, g\rangle = \mathrm{Cov}(f(H_2), g(Y)) = \mathrm{Cov}((K_\Phi f)(H_1), g(Y)) = \langle K_\Phi f,\, C^1_{HY}\, g\rangle = \langle f,\, K_\Phi^*\, C^1_{HY}\, g\rangle$, so $C^2_{HY} = K^*_\Phi\, C^1_{HY}$. By the dual Ky Fan inequality $\sigma_j(AB) \leq \|A\|_{\mathrm{op}}\, \sigma_j(B)$:

$$\sigma_k(C^2_{HY}) = \sigma_k(K^*_\Phi\, C^1_{HY}) \leq \|K^*_\Phi\|_{\mathrm{op}}\, \sigma_k(C^1_{HY}) = \|K_\Phi\|_{\mathrm{op}}\, \sigma_k(C^1_{HY}). \qquad \square$$

Remark 5.4 (Canonical correlations are invariant)*.* The singular values of C^i_{XH} are *not* canonical correlations; they are the unwhitened singular values. The canonical correlations (singular values of the whitened operator $\widetilde{C}^i_{XH} = C_{XX}^{-1/2}\, C^i_{XH}\, {C^i_{HH}}^{-1/2}$) do *not* satisfy a nontrivial DPI. Define $M = {C^1_{HH}}^{1/2}\, K_\Phi\, (K^*_\Phi\, C^1_{HH}\, K_\Phi)^{-1/2}$; then $M^*M = I$ and $\widetilde{C}^2_{XH} = \widetilde{C}^1_{XH}\, M$. When K_Φ is injective — the generic case for universal kernels on continuous hidden spaces — M is unitary, so $\sigma_j(\widetilde{C}^2_{XH}) = \sigma_j(\widetilde{C}^1_{XH})$ for all j. Canonical correlations are *invariant* under deterministic transitions.

This is a subtler manifestation of the L^2 isometry obstruction (§2.1): the whitening by C_{HH} perfectly compensates for the composition, leaving the canonical correlations unchanged. The unwhitened singular values are therefore the correct spectral quantities for detecting information loss across layers — they are sensitive to the RKHS-level complexity reduction that the canonical correlations absorb.

Remark 5.5 (When is K_Φ a contraction?)*.* The RKHS norm $\|g\|_{\mathcal{H}H}$ measures function complexity relative to the kernel. The condition $\|K_\Phi\|_{\mathrm{op}} > 1$ means that composing with Φ can increase this complexity: if Φ compresses part of its domain, the pullback $g \circ \Phi$ develops sharper variation in the compressed directions, inflating the RKHS norm. This is not an information increase — it is a kernel-architecture mismatch.

For a translation-invariant kernel $k_H(h_1, h_2) = m(\|h_1 - h_2\|)$ and a Lipschitz map Φ with constant $L \leq 1$: $\|K_\Phi g\|^2_{\mathcal{H}_H}$ is controlled by L and the kernel's smoothness. For smooth activations (GELU, SiLU) with Lipschitz constant ≤ 1, the RKHS composition operator is contractive under Matérn kernels with sufficient smoothness. ReLU requires separate treatment due to its non-smoothness: RKHS contractivity then depends on the kernel's regularity class. For a full network layer $\Phi(h) = \sigma(Wh + b)$, the affine part contributes $\|W\|_{\text{op}}$ to $\|K_\Phi\|_{\text{op}}$; layers with spectral normalization ($\|W\|_{\text{op}} = 1$) satisfy the contraction condition.

The contraction hypothesis should be verified for a given architecture, or enforced by choosing a kernel adapted to the computation (e.g., a wider bandwidth or the empirical neural tangent kernel). The quantitative form (24) is useful even when $\|K_\Phi\|_{\text{op}} > 1$: it bounds the rate at which unwhitened singular values can grow per layer.

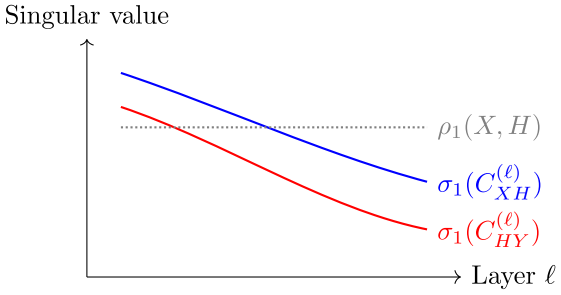
Spectral decay and routing compensation

Under the contraction hypothesis, the DPI (Theorem 5.3) implies that *both* the encoder-side and decoder-side unwhitened singular values decrease with depth:

- $\sigma_j(C^{(\ell)}_{XH})$ decreases — the hidden state decouples from the input as the encoder grows longer.
- $\sigma_k(C^{(\ell)}_{HY})$ also decreases — despite the shorter path from H to Y, the hidden state at a deeper layer carries less RKHS-level content, and this content loss dominates the shorter decoding path.

Meanwhile, the canonical correlations (whitened singular values) are invariant (Remark 5.4), and the end-to-end operator $\widetilde{C}_{XY} = \widetilde{C}_{XH}\,\widetilde{C}_{HY}$ is fixed. Since $\widetilde{C}^{(\ell)}_{XH} = \widetilde{C}^{(1)}_{XH}\, M_\ell$ for a unitary M_ℓ, the spectral content is preserved at the whitened level; all structural change is absorbed by the *routing matrix* $R^{(\ell)} = V^{(\ell)*}W^{(\ell)}$, which rotates to maintain the end-to-end factorization.

The interesting evolution across layers is therefore not a spectral seesaw between opposing magnitudes, but a *rotation of the routing geometry*: the directions along which information flows from input to output reorganize at each layer, even as the total spectral content (measured by canonical correlations) remains constant.



Unwhitened SVs decay; canonical correlations invariant

Figure 11: Spectral evolution across layers. Unwhitened singular values of both C_{XH} (blue) and C_{HY} (red) decrease with depth under the contraction hypothesis. Canonical correlations (dotted) are invariant for injective transitions. The routing matrix $R^{(\ell)}$ absorbs all structural change.

5.3 Estimation and sampling efficiency

The algorithm

Given N samples $\{x_n\}_{n=1}^N$ from a distribution on X:

1. Compute hidden activations $h_n = \psi(x_n)$ and outputs $y_n = \varphi(h_n)$.

2. Form centered Gram matrices:

$$\hat{G}^X_{nm} = k_X(x_n, x_m), \quad \hat{G}^H_{nm} = k_H(h_n, h_m), \quad \hat{G}^Y_{nm} = k_Y(y_n, y_m),$$

   then center each: $G \leftarrow G - \mathbf{1}_N G - G\mathbf{1}_N + \mathbf{1}_N G \mathbf{1}_N$ where $\mathbf{1}_N = \frac{1}{N}\mathbf{1}\mathbf{1}^\top$.

3. Solve the regularized kernel CCA problem for the encoder by computing the SVD of the matrix

$$M = (G^X + \lambda I)^{-1/2}\, G^X\, G^H\, (G^H + \lambda I)^{-1/2}. \tag{26}$$

   The singular values of M are the empirical canonical correlations $\hat{\sigma}_j$, and the left and right singular vectors give the coefficient vectors $\hat{\alpha}^{(\psi)}_j \in \mathbb{R}^N$ (after un-whitening by the appropriate inverse square roots).

4. Repeat for the decoder: kernel CCA between H and Y.

5. Compute the empirical routing matrix:

$$\hat{R}_{jk} = (\hat{\alpha}^{(\psi)}_j)^\top G^H\, \hat{\alpha}^{(\varphi)}_k. \tag{27}$$

   In theory, with unit-norm CCA vectors this equals the RKHS inner product $\langle w_j, \tilde{w}_k\rangle_{\mathcal{H}_H}$. In practice, regularization causes the empirical CCA vectors to have non-unit RKHS norm. The companion notebook uses a cosine normalization $\hat{R}_{jk}/(\|\hat{w}_j\|_{\mathcal{H}_H}\, \|\hat{\tilde{w}}_k\|_{\mathcal{H}_H})$ to enforce entries in $[-1, 1]$.

Computational cost: $O(N^3)$ for the eigendecompositions, $O(N^2 d)$ for forming Gram matrices. For large N, Nyström approximation reduces this to $O(Nm^2)$ where $m \ll N$ is the number of landmark points.

Estimation rates

Proposition 5.6 (Cross-covariance estimation; Fukumizu et al. [4])**.** *Under sub-Gaussian assumptions on the feature maps:*

$$\|\hat{C}_{XH} - C_{XH}\|_{\mathrm{HS}} = O(N^{-1/2}).$$

This is the standard parametric rate, achievable when the kernel is bounded.

Proposition 5.7 (Mode perturbation; Davis-Kahan [19])**.** *The angular error in the j-th singular vector satisfies*

$$\sin\angle(\hat{u}_j, u_j) \lesssim \frac{\|\hat{C}_{XH} - C_{XH}\|_{\mathrm{op}}}{\sigma_j - \sigma_{j+1}}. \tag{28}$$

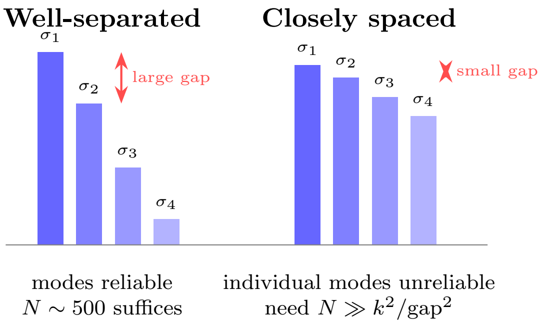


Figure 12: Effect of spectral gaps on mode estimation reliability. Well-separated singular values yield reliable individual modes with modest sample sizes; closely spaced singular values require $N \gg k^2/\text{gap}^2$ samples for accurate mode estimation.

The denominator — the *spectral gap* — is critical. Closely spaced singular values produce unreliable mode estimates.

This is the fundamental bottleneck for the routing matrix:

Corollary 5.8 (Routing matrix estimation error)**.**

$$|\hat{R}_{jk} - R_{jk}| \lesssim \frac{\|E_\psi\|}{\text{gap}_j^\psi} + \frac{\|E_\varphi\|}{\text{gap}_k^\varphi}$$

where E_ψ, E_φ *are estimation errors in the cross-covariance operators and* $\text{gap}_j = \sigma_j - \sigma_{j+1}$.

Proof. See Appendix A. □

6 Discussion

The routing matrix does not replace empirical interpretability work (probing, patching, circuit analysis). Its contribution is a formal language for stating and verifying interpretability claims: the informal "this representation encodes concept X for behavior Y" has a formal counterpart — "encoder mode j has routing weight R_{jk} to decoder mode k, estimated to precision $\pm\epsilon$ with N samples" — complete with gauge invariance, optimality guarantees, and estimation bounds. Whether these formal claims are practically useful depends on whether kernel CCA modes correspond to semantically meaningful features, which is an open empirical question.

6.1 Limitations

- **Kernel dependence.** The routing matrix depends on the kernel choice. This is a feature (different kernels probe different aspects) but means results must be interpreted relative to the kernel. Future work on multiscale analysis (sweeping over kernel bandwidths and identifying persistent features) would address this.

- **Skip connections.** The Markov condition $X \perp Y \mid H$ fails when Y depends on X through paths that bypass H (see Remark in §2.5). For residual networks, the routing matrix at each layer captures the incremental contribution $g_\ell(x)$ in $f^{(\ell)}(x) = x + g_\ell(x)$, not the full information flow.

- **Spectral gaps.** When canonical correlations are closely spaced, individual modes and routing entries are unreliable. The *aggregate* properties (total spectral mass, block structure of R) are more robust than individual entries.

- **Sampling cost.** Reliable estimation requires $N \gg k^2/\mathrm{gap}^2$, which can be large for fine-grained spectral structure.

- **Untested on language models.** The framework has been validated on toy systems (linear maps, small MLPs, finite-state transducers). Whether routing matrices reveal interpretable structure in transformer language models — where skip connections violate the Markov assumption and behavioral observables are harder to define — is an open empirical question.

- **Cross-kernel comparison.** Routing matrices computed under different kernels live in different mode bases and cannot be compared entry-wise, but the spectral profiles (σ_j) are well-defined scalars that can be directly compared across kernel choices (Proposition 3.9).

6.2 Extensions

Algorithmic comparison. The routing matrix at a single layer captures the *spatial wiring* of information at that point in the computation. But algorithmic identity involves *temporal* structure: two algorithms may wire information differently at each step while computing the same function. The signature for comparing algorithms is therefore the *sequence* of routing matrices across layers or timesteps: $(R^{(1)}, \ldots, R^{(L)})$, together with the spectral profiles $(\boldsymbol{\sigma}^{(\ell)}, \boldsymbol{\tau}^{(\ell)})$.

Comparing two such sequences — potentially of different lengths, with different hidden dimensions at each step — is a genuine open problem. The spectral DPI (§5.2) constrains the evolution of the unwhitened spectral profiles, but the routing matrices can change freely subject to the constraint that $\widetilde{C}_{XY}$ is constant. A proper metric on the space of such sequences would need to solve a temporal alignment problem (analogous to dynamic time warping) jointly with the mode-alignment problem at each layer. We leave this as future work, noting only that the single-layer routing distance $\|R^{(1)} - R^{(2)}\|_F$ is a necessary but far from sufficient condition for algorithmic equivalence — it captures wiring similarity at a snapshot but not the temporal organization that distinguishes algorithms.

The FST example (§B, Example B.3) illustrates the issue: the right-to-left carry propagation and two-pass algorithms differ in *when* they process information. Their single-step routing matrices may look similar at some timesteps and radically different at others; the diagnostic is in the full profile, not any single slice.

A Proofs

Proof of Theorem 5.1 (Factored truncation bounds). **Part (a).** Write $\widetilde{C}_{XY} - \widehat{T} = (\widetilde{C}_{XH} - (\widetilde{C}_{XH})_{k_1})\,\widetilde{C}_{HY} + (\widetilde{C}_{XH})_{k_1}\,(\widetilde{C}_{HY} - (\widetilde{C}_{HY})_{k_2})$ (adding and subtracting $(\widetilde{C}_{XH})_{k_1}\,\widetilde{C}_{HY}$).

By submultiplicativity of the operator norm:

$$\begin{aligned}\|\widetilde{C}_{XY} - \widehat{T}\|_{\mathrm{op}} &\le \|\widetilde{C}_{XH} - (\widetilde{C}_{XH})_{k_1}\|_{\mathrm{op}}\,\|\widetilde{C}_{HY}\|_{\mathrm{op}} + \|(\widetilde{C}_{XH})_{k_1}\|_{\mathrm{op}}\,\|\widetilde{C}_{HY} - (\widetilde{C}_{HY})_{k_2}\|_{\mathrm{op}} && (29)\\ &= \sigma_{k_1+1}\,\tau_1 + \sigma_1\,\tau_{k_2+1}. && (30)\end{aligned}$$

Part (b). From (13), $\widetilde{C}_{XY} = \sum_{j,k} \sigma_j R_{jk} \tau_k \, u_j \otimes v_k$. The truncated product retains only terms with $j \le k_1$ and $k \le k_2$. Since $\{u_j \otimes v_k\}$ are orthonormal in the HS inner product:

$$\|\widetilde{C}_{XY} - \widehat{T}\|_{\mathrm{HS}}^2 = \sum_{\substack{j>k_1 \\ \text{or } k>k_2}} (\sigma_j R_{jk} \tau_k)^2. \qquad \square$$

Proof of Corollary 5.8 (Routing matrix estimation error). The error in $R_{jk} = \langle w_j, \tilde{w}_k \rangle$ comes from errors in both singular vectors. Since w_j, $\tilde{w}_k$, $\hat{w}_j$, $\hat{\tilde{w}}_k$ all have unit RKHS norm (they are singular vectors of whitened operators): $|\langle \hat{w}_j, \hat{\tilde{w}}_k \rangle - \langle w_j, \tilde{w}_k \rangle| \le \|\hat{w}_j - w_j\| \, \|\hat{\tilde{w}}_k\| + \|w_j\| \, \|\hat{\tilde{w}}_k - \tilde{w}_k\| \le \|\hat{w}_j - w_j\| + \|\hat{\tilde{w}}_k - \tilde{w}_k\|$. Bounding each term via Davis-Kahan gives the stated bound. (In the empirical setting with regularization, the CCA vectors require renormalization; see §5.3.) Here we use the convention that $\hat{w}_j$ is chosen with the same sign orientation as w_j (i.e., $\langle \hat{w}_j, w_j \rangle \ge 0$), which is standard in perturbation theory. Singular vectors are defined only up to sign, and without this convention the error bound applies to $\min_{\epsilon \in \{-1,1\}} |\hat{R}_{jk} - \epsilon \, R_{jk}|$. $\square$

B Worked Examples

B.1 Example 1: Linear systems (closed form)

Example B.1 (Linear factorization with linear kernel)**.** Let $\psi(x) = Ax$ and $\varphi(h) = Bh$ with $A \in \mathbb{R}^{d_H \times d_X}$, $B \in \mathbb{R}^{d_Y \times d_H}$, and $x \sim \mathcal{N}(0, \Sigma_X)$. Use the linear kernel $k(u,v) = u^\top v$ on all spaces, so $\mathcal{H}_X = \mathbb{R}^{d_X}$ etc.

The cross-covariance matrices are:

$$C_{XH} = \mathrm{Cov}(x, Ax) = \Sigma_X A^\top, \tag{31}$$

$$C_{HY} = \mathrm{Cov}(Ax, BAx) = A \Sigma_X A^\top B^\top. \tag{32}$$

The whitened encoder operator is:

$$\widetilde{C}_{XH} = \Sigma_X^{-1/2} \, \Sigma_X A^\top \, (A \Sigma_X A^\top)^{-1/2} = \Sigma_X^{1/2} A^\top (A \Sigma_X A^\top)^{-1/2}.$$

Its SVD gives the canonical correlations between x and Ax and the canonical directions.

Comparing two linear systems: Take $A_1 \in \mathbb{R}^{8 \times 5}$, $B_1 \in \mathbb{R}^{3 \times 8}$ and $A_2 \in \mathbb{R}^{4 \times 5}$, $B_2 \in \mathbb{R}^{3 \times 4}$ with $B_1 A_1 \approx B_2 A_2$ (same end-to-end map, different hidden dimensions).

System 1 has a rank-8 hidden state routing through $R^1 \in \mathbb{R}^{5 \times 3}$ (at most); System 2 has a rank-4 hidden state with $R^2 \in \mathbb{R}^{4 \times 3}$. The routing matrices will generically differ even though $d_{\mathrm{IO}} \approx 0$. The wider system has more modes available and can spread information across them; the narrower system is forced to concentrate.

B.2 Example 2: Two-layer MLPs

Example B.2 (Nonlinear comparison)**.** Train two MLPs on the regression task $y = \sin(x_1) + \frac{1}{2}\cos(2x_2) + \frac{3}{10}\sin(3x_3)$ with $x \in \mathbb{R}^3$:

- **MLP A:** hidden dimension 32, ReLU activation.
- **MLP B:** hidden dimension 8, ReLU activation.

Both achieve similar test loss ($d_{\mathrm{IO}} \approx 0$). Using RBF kernels with median-heuristic bandwidth and $N = 400$ test samples, we compute:

1. The encoder canonical correlations σ_j^A and σ_j^B. MLP A has a slower spectral decay (more modes carrying information, as expected from the wider hidden layer).

2. The routing matrices R^A and R^B. MLP A's routing matrix is more nearly diagonal (the wide hidden layer can afford serial routing), while MLP B's has more off-diagonal structure (the narrow bottleneck forces mode mixing).

These predictions are verified in the companion notebook `routing-matrix-experiments.py`.

B.3 Example 3: Finite-state transducers

Example B.3 (Discrete computation)**.** The framework applies to any sequential computation, not just neural networks. Consider two deterministic finite-state transducers (FSTs) that both compute binary increment (e.g., $0110 \mapsto 0111$, $0111 \mapsto 1000$):

- **FST A** (right-to-left carry propagation): Scans from the least significant bit, flipping bits until it finds a 0. States encode "carrying" vs. "done."
- **FST B** (two-pass): First pass marks the rightmost 0; second pass flips all bits between the mark and the LSB. States encode "scanning," "marked," "flipping," and "done."

Both compute the same function ($d_{\text{IO}} = 0$), but their intermediate state sequences differ.

To apply the framework: let $X = Y = \{0, 1\}^n$ (binary strings of length n), and let $H^{(t)}$ be the FST's configuration (state, head position, tape contents) at step t. Define kernels on these discrete spaces — for instance, the *Hamming kernel* $k(s, s') = \exp(-d_{\text{Ham}}(s, s')/\sigma)$ on binary strings, or simply the inner product of one-hot encodings on the finite state set.

The routing matrix at step t reveals how input features (which input bits the state has "seen") connect to output features (which output bits are determined) through the internal state. For FST A, the routing is sequential: at step t, the state has processed bits $1, \ldots, t$ and determined output bits $1, \ldots, t$. The routing matrix is nearly diagonal. For FST B, the first pass reads all bits but produces no output; the second pass then produces all outputs. The routing matrix during the first pass has large input-side correlations but small output-side correlations; this reverses in the second pass — the routing geometry rotates within a single computation.

This illustrates a general point: the routing matrix distinguishes *algorithms*, not just *functions.* Any comparison method based solely on input-output behavior (including CKA applied to I/O pairs) would rate these FSTs as identical. The routing matrix sees the difference.

C Connections to Other Frameworks

C.1 Linear systems theory

A linear time-invariant system with realization (A, B, C) has Hankel operator $\mathcal{H} = \mathcal{O}\mathcal{R}$ where $\mathcal{O}$ (observability) and $\mathcal{R}$ (reachability) play the roles of $\widetilde{C}_{HY}$ and $\widetilde{C}_{XH}^*$ (Moore [9]). The Hankel singular values are the singular values of $\mathcal{H}$, and balanced truncation keeps the top k (Glover [5]). The routing matrix generalizes the Hankel matrix to nonlinear maps; the factored Eckart-Young theorem (§5) generalizes balanced truncation error bounds.

C.2 Quantum information theory

The Choi-Jamiołkowski isomorphism (see Wilde [13]) maps a quantum channel Φ to a bipartite state J_Φ, whose Schmidt decomposition gives the channel's principal axes. The routing matrix plays an analogous role: it decomposes the "channel" $X \to H \to Y$ into principal information-routing axes. The spectral DPI (Theorem 5.3) is the classical analogue of the quantum data processing inequality, with RKHS regularity playing the role of quantum noise in breaking the isometry.

C.3 CKA and representational similarity

CKA (Kornblith et al. [6]) computes a normalized similarity between two hidden representations. In our framework, CKA corresponds to the normalized inner product of the hidden-state Gram matrices, which aggregates over all spectral, alignment, and routing information into a single scalar. The three-level decomposition (§4.3) refines CKA into interpretable components.

C.4 Spectral learning of weighted finite automata

Spectral learning algorithms for weighted finite automata (WFAs) recover a minimal state representation from a Hankel matrix of observed input-output sequences (Hsu, Kakade & Zhang [14]; Balle et al. [16]). The core step is identical to ours: factor the Hankel matrix via SVD into an "observation" part and a "system" part, then read off the transition operators in the SVD basis.

Concretely, spectral WFA learning computes the SVD $H = U\Sigma V^\top$ of the finite Hankel matrix (whose rows are indexed by prefixes and columns by suffixes), then defines the learned forward and backward operators in the truncated SVD basis. In our framework, the cross-covariance operator $C_{XY} = C_{XH}\, C_{HH}^{-1}\, C_{HY}$ plays the role of the Hankel matrix, and the routing decomposition $\widetilde{C}_{XY} = \sum_{j,k} \sigma_j R_{jk} \tau_k \, u_j \otimes v_k$ is the analogue of the SVD-based factorization.

The routing matrix framework extends spectral WFA learning in two directions:

1. **From finite to continuous.** Spectral WFA learning operates on finite Hankel matrices over a discrete alphabet. The kernel version replaces these with cross-covariance operators on RKHSs, handling continuous state spaces without discretization.
2. **From linear to nonlinear.** WFA learning assumes the state-update and output maps are bilinear in the spectral basis (the WFA structure). The kernel framework makes no such assumption: the encoder and decoder can be arbitrary nonlinear maps, with the kernel controlling which aspects of the nonlinearity are resolved.

The estimation theory also connects: spectral WFA learning requires a spectral gap in the Hankel matrix for the learned automaton to be close to the true one (Hsu et al. [14], Theorem 3), exactly paralleling our Davis-Kahan-based bounds (Corollary 5.8).

C.5 The information bottleneck

The information bottleneck (IB) method (Tishby, Pereira & Bialek [17]) seeks a compressed representation T of input X that retains maximal information about output Y:

$$\min_{p(t|x)} \; I(X;T) - \beta\, I(T;Y).$$

The IB framework and the routing matrix both decompose a factored computation $X \to H \to Y$ into input-relevant and output-relevant components. The routing framework sidesteps several known difficulties with the IB:

1. **No differential entropy estimation.** The IB requires estimating mutual information $I(X;H)$ on continuous spaces, which involves density ratios in high dimensions. State-of-the-art MI estimators (MINE, variational bounds) have high variance and are sensitive to hyperparameters. Saxe et al. [18] showed that the apparent "compression phase" in deep learning reported by Shwartz-Ziv & Tishby (2017) was an artifact of binning: replacing the binned MI estimator with a more careful one eliminated the compression. The routing framework replaces MI estimation with kernel CCA — an eigenvalue problem on Gram matrices — which has well-understood convergence rates (§5.3) and no binning.

2. **Structured decomposition vs. scalar tradeoff.** The IB produces a scalar tradeoff curve $I(X;T)$ vs. $I(T;Y)$ parameterized by β. The routing matrix provides a *structured* decomposition: per-mode canonical correlations σ_j (input-side) and τ_k (output-side) with the routing matrix R_{jk} connecting them. This is strictly more informative: the spectral profiles contain the same ordering information as the IB curve (which modes carry most input-information, which are most output-relevant), while the routing matrix adds the wiring structure.

3. **RKHS geometry vs. distributional assumptions.** The IB is defined for arbitrary joint distributions but in practice requires either discrete variables or density estimation. The kernel framework works directly with RKHS inner products, which are computable from kernel evaluations without density estimation. The kernel choice determines which features are visible (analogous to the IB's choice of what "information" means). This trades one sensitivity for another: the IB is sensitive to the MI estimator, while the routing framework is sensitive to the kernel choice. The advantage is that kernel dependence is explicit (a hyperparameter that can be varied and compared; see §6) rather than an artifact of an estimation procedure.

The connection can be made precise via Bach's kernel mutual information [3]: the kernel MI between X and H is $I_k(X;H) = -\frac{1}{2}\sum_j \log(1-\sigma_j^2)$, where σ_j are the canonical correlations. This is a monotone function of the spectral profile $\{\sigma_j\}$, so the kernel CCA spectrum contains the same information as the kernel MI. The routing matrix adds the factored structure that the IB discards.

D Kernelized Computational Mechanics

The framework developed in this report, when applied to the factorization Past → State → Future of a stochastic process, yields a kernel generalization of computational mechanics (Crutchfield & Young [1]). This connection is not merely an analogy: the objects are the same, viewed through the kernel lens.

D.1 Causal states and the Hankel matrix

In computational mechanics, the *causal states* of a process are equivalence classes of pasts that give the same predictive distribution over futures:

$$x_{\leq t} \sim x'_{\leq t} \quad \Longleftrightarrow \quad P(x_{>t} \mid x_{\leq t}) = P(x_{>t} \mid x'_{\leq t}).$$

The resulting minimal sufficient statistic is the *epsilon-machine*, and the number of causal states is the *statistical complexity* C_μ.

The central object in computational mechanics is the *Hankel matrix* $H_{ij} = P(\text{future}_i, \text{past}_j)$, whose rank equals the number of causal states. This is the (uncentered) cross-covariance matrix between past and future indicators.

D.2 The kernel generalization

Replace distributional equality with RKHS equality: two pasts are kernel-equivalent iff their conditional embeddings of futures coincide:

$$x_{\leq t} \sim_k x'_{\leq t} \quad \Longleftrightarrow \quad \mu_{F|P=x_{\leq t}} = \mu_{F|P=x'_{\leq t}} \quad \text{in } \mathcal{H}_F,$$

where $\mu_{F|P}$ is the kernel conditional mean embedding (Song et al. [11]).

The *kernel Hankel operator* is the cross-covariance operator $C_{\text{Past,Future}} \colon \mathcal{H}_F \to \mathcal{H}_P$. Its SVD gives:

- **Right singular vectors** (in $\mathcal{H}_F$): the kernel predictive features — the most predictable aspects of the future.
- **Left singular vectors** (in $\mathcal{H}_P$): the kernel causal states — the past features most relevant for prediction.
- **Singular values**: the strength of each predictive mode — how much each causal-state direction contributes to prediction.

D.3 Routing as transition structure

In classical computational mechanics, the epsilon-machine has transition probabilities between causal states. Now consider two time slices of a process:

$$\text{Past} \xrightarrow{\psi_1} \text{State}_{t_1} \xrightarrow{\Phi} \text{State}_{t_2} \xrightarrow{\varphi_2} \text{Future}.$$

The routing matrix R_{jk} between the two slices captures how the causal-state modes at t_1 connect to those at t_2. This is the kernel generalization of the epsilon-machine's transition matrix:

- Diagonal R: causal states are preserved across the transition (the process has stable structure).
- Off-diagonal R: causal states mix (the process reorganizes its predictive structure).
- The spectral DPI constrains how the predictive mode strengths evolve: they can only decrease through the transition (information about the past is lost).

D.4 Kernel statistical complexity

Bach's kernel mutual information [3] gives the kernel analogue of the excess entropy (predictive information):

$$I_k(\text{Past; Future}) = -\tfrac{1}{2}\sum_i \log(1 - \sigma_i^2),$$

where σ_i are the canonical correlations between past and future. This decomposes into per-mode contributions — something classical computational mechanics cannot do without choosing a basis for the causal-state space. The effective number of modes with σ_i above a threshold gives a kernel analogue of the statistical complexity C_μ.

The connection is summarized:

Computational mechanics	**Kernel version (this framework)**
Hankel matrix H_{ij}	Cross-covariance C_{PF}
Causal states	Left singular vectors of C_{PF}
Predictive features	Right singular vectors of C_{PF}
$\operatorname{rank}(H) = \#$ causal states	Effective rank of C_{PF}
Transition probabilities	Routing matrix R_{jk}
Excess entropy E	Kernel MI: $-\frac{1}{2}\sum \log(1-\sigma_i^2)$
Statistical complexity C_μ	Effective number of modes above threshold

D.5 Proofs of the correspondences

We now prove the key claims in the table above.

Proposition D.1 (Kernel Hankel = cross-covariance)**.** *Let $(X_t)_{t\in\mathbb{Z}}$ be a stationary stochastic process. Define $P = (X_t, X_{t-1}, \dots)$ (past) and $F = (X_{t+1}, X_{t+2}, \dots)$ (future), with kernels k_P, k_F on the past and future sequence spaces. Then the kernel Hankel operator — the operator whose matrix elements in a basis of past/future indicator functions reproduce the classical Hankel matrix — is exactly the cross-covariance operator $C_{PF}\colon \mathcal{H}_F \to \mathcal{H}_P$.*

Proof. The classical Hankel matrix has entries $H_{ij} = P(\text{future}_i \cap \text{past}_j)$. For indicator kernels $k_P(p,p') = \mathbf{1}[p = p']$ and similarly for futures, the RKHS inner product is the Kronecker delta, and the cross-covariance evaluates as

$$\langle k_P(\cdot, p_j),\, C_{PF}\, k_F(\cdot, f_i)\rangle_P = \mathbb{E}[k_P(P, p_j)\, k_F(F, f_i)] = P(P = p_j,\, F = f_i) = H_{ij}.$$

For general kernels, C_{PF} replaces the discrete past-future joint probability with the RKHS-valued cross-covariance, which captures all dependencies detectable by k_P and k_F. The SVD of C_{PF} generalizes the SVD of H — the classical basis for identifying causal states via the rank of H. □

Proposition D.2 (Routing matrix as kernel transition matrix)**.** *Consider the factorization Past $\xrightarrow{\psi_1} S_{t_1} \xrightarrow{\Phi} S_{t_2} \xrightarrow{\varphi_2}$ Future, where S_{t_i} are causal-state representations at two times. The routing matrix $R_{jk} = \langle w_j, \tilde{w}_k\rangle_{\mathcal{H}_S}$ between the encoder modes at t_1 and decoder modes at t_2 is the kernel generalization of the transition probability matrix $T_{jk} = P(S_{t_2} = s_k \mid S_{t_1} = s_j)$ of the epsilon-machine.*

Proof. In classical computational mechanics with discrete causal states, the encoder modes are indicator functions $w_j = \mathbf{1}[S = s_j]$ (orthonormal under the counting kernel) and similarly for decoder modes. The routing matrix entry is

$$R_{jk} = \langle \mathbf{1}[S_{t_1} = s_j],\, \mathbf{1}[S_{t_2} = s_k]\rangle_{\mathcal{H}_S} = \mathbb{E}[\mathbf{1}[S_{t_1} = s_j]\, \mathbf{1}[S_{t_2} = s_k]]/\sqrt{P(s_j)\, P(s_k)},$$

which is a normalized version of the joint probability — the same information as the transition matrix T_{jk} (up to the stationary distribution weighting).

For general kernels, w_j and $\tilde{w}_k$ are RKHS functions (nonlinear features of the state space), and R_{jk} captures the overlap between the j-th predictive mode at t_1 and the k-th predictive mode at t_2. The routing matrix thus generalizes the transition matrix from a matrix over discrete states to one over continuous RKHS modes. □

Proposition D.3 (Kernel statistical complexity)**.** *Let $\sigma_1 \geq \sigma_2 \geq \cdots$ be the canonical correlations between past and future. The kernel excess entropy is*

$$E_k = -\tfrac{1}{2} \sum_i \log(1 - \sigma_i^2),$$

and the kernel statistical complexity (effective number of causal modes) is

$$C_k(\epsilon) = |\{i : \sigma_i > \epsilon\}|,$$

where ϵ is a significance threshold determined by the estimation error bounds of §5.3.

Proof. The excess entropy formula follows from Bach's kernel mutual information [3]: for Gaussian processes, the mutual information between two random variables equals $-\frac{1}{2}\sum_i \log(1-\rho_i^2)$ where ρ_i are the canonical correlations. Bach shows this extends to the RKHS setting. Each term $-\frac{1}{2}\log(1-\sigma_i^2)$ is the information contributed by the i-th predictive mode — a per-mode decomposition that classical computational mechanics cannot provide without an arbitrary basis choice.

The classical statistical complexity $C_\mu = \log|\mathcal{S}|$ (for an epsilon-machine with state set $\mathcal{S}$) counts the number of distinct causal states. In the kernel setting, the "number of causal states" is infinite (continuous state space), but the effective dimensionality of the predictive subspace is captured by the number of canonical correlations above the noise floor. The threshold ϵ should be chosen based on the Davis-Kahan bound: modes with $\sigma_i < O(N^{-1/2}/\mathrm{gap}_i)$ are statistically indistinguishable from noise. □

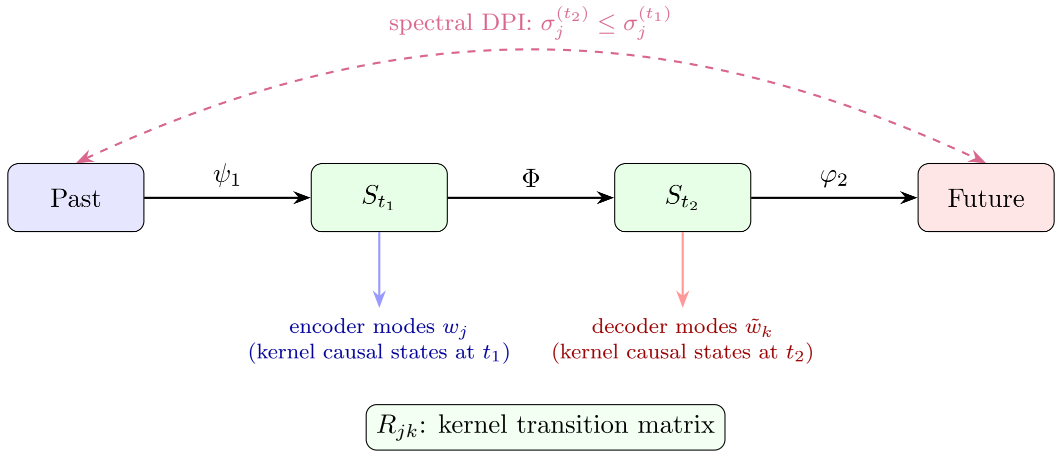

spectral DPI: $\sigma_j^{(t_2)} \leq \sigma_j^{(t_1)}$
Past
ψ_1
S_{t_1}
Φ
S_{t_2}
φ_2
Future
encoder modes w_j
(kernel causal states at t_1)
decoder modes $\tilde{w}_k$
(kernel causal states at t_2)
R_{jk}: kernel transition matrix

Figure 13: Temporal routing in computational mechanics. The routing matrix R_{jk} between kernel causal states at times t_1 and t_2 acts as a kernel transition matrix; the spectral DPI guarantees that predictive canonical correlations can only decrease over time.

D.6 The spectral DPI as information loss in the epsilon-machine

The spectral DPI (Theorem 5.3) applied to the computational mechanics setting says: as the process evolves from t_1 to t_2, the canonical correlations between the current state and the future can only decrease:

$$\sigma_j^{(t_2)} \leq \sigma_j^{(t_1)}, \quad \text{for all } j.$$

In epsilon-machine language, this means each predictive mode can only *lose* information about the future as the process evolves — the epsilon-machine's predictive power is maximal at the present and decays into the past. This is the kernel analogue of the classical result that the epsilon-machine is the *minimal* sufficient statistic for future prediction.

The routing matrix R_{jk} between time slices determines which predictive modes survive the transition and which are lost. A diagonal R with entries close to 1 means the causal-state structure is stable (a near-stationary process); off-diagonal structure or small diagonal entries indicate reorganization of the predictive modes (regime changes).

D.7 Worked example: a two-state hidden Markov model

Consider a binary HMM with two hidden states $s \in \{0, 1\}$, symmetric transition probability $T_{01} = T_{10} = p$, and emission probabilities $P(X_t = 1 \mid S_t = 0) = \epsilon$, $P(X_t = 1 \mid S_t = 1) = 1 - \epsilon$.

Classical analysis. The epsilon-machine has two causal states (the hidden states themselves, assuming $\epsilon \neq 1/2$). The transition matrix is $T = \left(\begin{smallmatrix} 1-p & p \\ p & 1-p \end{smallmatrix}\right)$, with eigenvalues 1 and $1 - 2p$. The statistical complexity is $C_\mu = \log 2 \approx 0.693$ nats.

Kernel analysis. With a linear kernel on past windows of length L, the cross-covariance operator C_{PF} has rank at most 2 (since there are only two causal states). The top canonical correlation is

$$\sigma_1 = (1 - 2\epsilon)^2 (1 - 2p)^{L-1}.$$

Derivation: The past-future cross-covariance factors as $C_{PF} = C_{PS}\, T\, C_{SF}$, where C_{PS} and C_{SF} are the past-to-state and state-to-future cross-covariances. The factor $(1 - 2\epsilon)^2$ is the squared emission distinguishability (the canonical correlation between a single observation and the hidden state for the binary symmetric channel), and $(1 - 2p)^{L-1}$ is the L-step mixing decay (the second eigenvalue of T raised to the power $L - 1$). The second canonical correlation $\sigma_2 = 0$ for windows longer than the mixing time (rank of C_{PF} is at most 2).

The routing matrix between two time slices separated by δ steps is

$$R = \begin{pmatrix} 1 - p\delta & p\delta \\ p\delta & 1 - p\delta \end{pmatrix} + O(\delta^2).$$

This follows from Proposition D.2: for indicator kernels on the two-state space, the routing matrix entries R_{jk} reduce to normalized joint probabilities, which for the symmetric two-state chain are the entries of $T^\delta = I + \delta(T - I) + O(\delta^2)$. The spectral DPI gives $\sigma_1^{(t+\delta)} = (1 - 2p)\, \sigma_1^{(t)}$, confirming that predictive information decays geometrically at rate $1 - 2p$ per time step.

For p close to 1/2 (rapidly mixing), the routing matrix quickly approaches the identity divided by 2 — all predictive structure is lost. For p close to 0 (slowly mixing), R remains nearly diagonal for many time steps — the causal-state structure is preserved.

Acknowledgments

I am grateful to Richard Ngo for mentorship and many conversations that shaped the direction of this work. Thanks to Maria for research management and keeping things on track. Thanks to my fellow MATS 9.0 mentees for feedback and camaraderie, and to the Lighthaven staff for providing a productive working environment. Finally, thanks to my friends and family for their support.

Written with assistance from Claude (Anthropic).

References

[1] Crutchfield, J. P. & Young, K. (1989). Inferring statistical complexity. *Physical Review Letters*, 63(2), 105–108.

[2] Bach, F. R. & Jordan, M. I. (2002). Kernel independent component analysis. *JMLR*, 3, 1–48.

[3] Bach, F. R. (2023). Information theory with kernel methods. *IEEE Trans. Inf. Theory*, 69(2), 752–775.

[4] Fukumizu, K., Bach, F. R., & Jordan, M. I. (2004). Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *JMLR*, 5, 73–99.

[5] Glover, K. (1984). All optimal Hankel-norm approximations of linear multivariable systems and their L^∞-error bounds. *Int. J. Control*, 39(6), 1115–1193.

[6] Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. *ICML*.

[7] Kriegeskorte, N., Mur, M., & Bandettini, P. (2008). Representational similarity analysis. *Frontiers in Systems Neuroscience*, 2, 4.

[8] Morcos, A. S., Raghu, M., & Bengio, S. (2018). Insights on representational similarity in neural networks with canonical correlation. *NeurIPS*.

[9] Moore, B. (1981). Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE TAC*, 26(1), 17–32.

[10] Raghu, M., Gilmer, J., Yosinski, J., & Sohl-Dickstein, J. (2017). SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *NeurIPS*.

[11] Song, L., Huang, J., Smola, A., & Fukumizu, K. (2009). Hilbert space embeddings of conditional distributions with applications to dynamical systems. *ICML*.

[12] Wendland, H. (2004). *Scattered Data Approximation*. Cambridge University Press.

[13] Wilde, M. M. (2017). *Quantum Information Theory* (2nd ed.). Cambridge University Press.

[14] Hsu, D., Kakade, S. M., & Zhang, T. (2012). A spectral algorithm for learning hidden Markov models. *JCSS*, 78(5), 1460–1480.

[15] Björck, Å. & Golub, G. H. (1973). Numerical methods for computing angles between linear subspaces. *Math. Comp.*, 27(123), 579–594.

[16] Balle, B., Carreras, X., Luque, F. M., & Quattoni, A. (2014). Spectral learning of weighted automata: a forward-backward perspective. *Machine Learning*, 96(1), 33–63.

[17] Tishby, N., Pereira, F. C., & Bialek, W. (1999). The information bottleneck method. *Proc. 37th Allerton Conf.*, 368–377.

[18] Saxe, A. M., Bansal, Y., Dapello, J., Advani, M., Kolchinsky, A., Tracey, B. D., & Cox, D. D. (2019). On the information bottleneck theory of deep learning. *JSTAT*, 2019(12), 124020.

[19] Davis, C. & Kahan, W. M. (1970). The rotation of eigenvectors by a perturbation. III. *SIAM J. Numer. Anal.*, 7(1), 1–46.

[20] Eckart, C. & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3), 211–218.

[21] Mirsky, L. (1960). Symmetric gauge functions and unitarily invariant norms. *Quart. J. Math.*, 11(1), 50–59.

[22] von Neumann, J. (1937). Some matrix-inequalities and metrization of matrix-space. *Tomsk Univ. Rev.*, 1, 286–300.

[23] Jordan, C. (1875). Essai sur la géométrie à n dimensions. *Bull. Soc. Math. France*, 3, 103–174.

[24] Gretton, A., Bousquet, O., Smola, A., & Schölkopf, B. (2005). Measuring statistical dependence with Hilbert-Schmidt norms. *ALT*, 63–77.

[25] Belghazi, M. I., Barber, A., Rajeshwar, S., et al. (2018). Mutual information neural estimation. *ICML*.

[26] Bhatia, R. (1997). *Matrix Analysis.* Springer.