A Categorical Systems-Theoretic Survey of Neural Network Architectures

Open Dynamical Systems, Wiring Diagrams, and (Pro)functorial Semantics

Leo Gorodinski
Cohesive Systems

February 27, 2026

Abstract

The term *neural architecture* is used inconsistently in machine learning, alternately referring to computation graphs (e.g. MLPs, CNNs, Transformers), interaction patterns between multiple inputs (e.g. bi-encoders versus cross-encoders), dynamical regimes (feedforward versus recurrent), and even objective-driven templates (e.g. autoencoders). This survey organizes a representative set of architectures using the language of categorical systems theory and *open dynamical systems*. Building on wiring diagram operads, (decorated/structured) cospans, and Markov categories, we show how many architectures can be viewed as morphisms in symmetric monoidal categories (or double categories) of open systems. Within this setting, convolution, message passing, and attention admit a common description as (possibly enriched) weighted colimits/coends against learned "kernel" data, i.e. profunctorial semantics. The goal is expository: to align common ML constructs with established categorical frameworks and to separate orthogonal axes of design in a precise way.

Contents

1 Introduction

Deep learning practice distinguishes *model families* (Transformers, CNNs, GNNs), *interaction patterns* (bi-encoder, cross-encoder), *dynamical regimes* (feedforward, recurrent, continuous-depth), and *training templates* (autoencoders, diffusion). However, these notions are often conflated under the umbrella term "architecture". This survey proposes a categorical organizing principle: treat neural models as *open systems* that can be wired together along typed interfaces, and view common layers as compositional constructions whose semantics is given by categorical colimits (often in enriched form).

The relevant mathematical infrastructure is now well developed in applied category theory: wiring diagram operads [Spi13], algebras of open dynamical systems [VSL15], cospan-based formalisms for open systems [Fon15; BC20; BCV22], black-boxing and hypergraph categories [BF18; Fon18], and probabilistic generalizations via Markov categories [Fri20; CJ19]. String diagrammatic reasoning, central to categorical quantum mechanics [CK17], has also influenced compositional treatments of meaning and representation learning [CSC10].

Scope. We focus on the architectures discussed in the surrounding conversation: MLPs and feedforward networks; CNNs; GNNs; Transformers (attention); residual networks; RNNs and continuous-depth models; bi-encoders and cross-encoders; (variational) autoencoders; diffusion-like Markovian generative models; and mixtures-of-experts. The emphasis is on a *classification* in categorical systems terms, not on new theorems.

Reader assumptions. We assume basic familiarity with monoidal categories and string diagrams, and with standard neural network constructions. For categorical background in an applied style, see [FS19].

2 Categorical systems theory and open dynamical systems

We recall the principal ingredients used to model "systems with interfaces" compositionally. Several equivalent-looking frameworks exist; part of their value is precisely to separate *syntax* (wiring patterns) from *semantics* (behavior).

2.1 Wiring diagrams and operadic syntax

Wiring diagrams describe how component systems connect. Spivak formalizes wiring diagrams as an operad whose operations encode hierarchical nesting and substitution [Spi13]; temporal variants

support discrete-time process composition [RS13]. Vagner–Spivak–Lerman develop algebras of open dynamical systems on such operads [VSL15].

Remark 2.1 (Operad versus category). Operads capture multi-input/multi-output substitution directly; symmetric monoidal categories can be recovered (e.g. as PROP-like envelopes) when one also wants sequential and parallel composition. In practice, ML "computation graphs" can be treated in either formalism; we use whichever is notationally clearer.

2.2 Cospans, decorated cospans, and structured cospans

A common categorical encoding of open systems uses cospans $X \to N \leftarrow Y$ in a base category (e.g. **FinSet**, topological spaces, manifolds), where X and Y are interfaces and N is the internal system.

Decorations add dynamics or additional structure at the apex. Fong's decorated cospans construction [Fon15] yields symmetric monoidal categories of open systems with chosen decoration functors; decorated corelations [Fon18] refine this to build hypergraph categories and semantic "black-box" functors. Baez and collaborators develop structured cospans as an alternative, and relate structured and decorated approaches [BC20; BCV22].

2.3 Open dynamical systems doctrine

Baez–Pollard treat open reaction networks as morphisms in a category and construct functors to open dynamical systems and to steady-state relations [BP17]. Myers develops a general account of open dynamical systems in a double categorical setting [Mye21] and a broader "systems doctrine" viewpoint in his draft book [Mye23].

For our purposes, it suffices to model (discrete-time) neural networks as open systems with typed ports. Continuous-time models (Neural ODEs) fit similarly, but in a different doctrine.

Definition 2.1 (Informal: open discrete-time dynamical system)**.** Fix a category $\mathcal{V}$ of state spaces (often **Vect** or smooth manifolds). An *open discrete-time system* with input space U and output space Y consists of a state space S and update/readout morphisms

$$S \times U \xrightarrow{f} S, \qquad S \xrightarrow{g} Y,$$

with composition given by wiring outputs to inputs (typically via pullbacks/pushouts depending on the doctrine). In neural networks, parameters are carried either as additional state or as decoration.

3 Mathematical semantics for "layers": kernels, profunctors, and coends

Many ML layers can be understood as computing an output representation by aggregating input representations using a learned "kernel" or "interaction rule". Categorical language provides a unifying description.

3.1 Data as diagrams over an index category

Let $\mathcal{I}$ encode the *shape* of an input: a finite set of tokens, a grid, a graph, or a time index. An input representation is a functor

$$H : \mathcal{I} \to \mathcal{V},$$

where $\mathcal{V}$ is a category of feature objects (often vector spaces or modules).

Examples of $\mathcal{I}$:

- sequences: a discrete category on positions $\{1, \dots, n\}$ or a thin category $(\{1, \dots, n\}, \leq)$ if causality/order matters;
- grids: a category carrying a $\mathbb{Z}^2$-action (translation symmetry) or a groupoid of patch locations;
- graphs: a small category generated by vertices and edges.

3.2 Profunctors as interaction kernels

A (Set-valued) profunctor is a functor $P : \mathcal{I}^{\mathrm{op}} \times \mathcal{J} \to \mathbf{Set}$. In enriched form [Kel82], profunctors can take values in a monoidal category $(\mathcal{W}, \otimes)$, for example nonnegative reals (as a quantale), probability distributions, or vector spaces. Intuitively, $P(i, j)$ is the "amount of influence" of j on i.

The bicategory $\mathbf{Prof}_{\mathcal{W}}$ composes profunctors by a coend:

$$(P \circ Q)(i, k) \;\cong\; \int^{j \in \mathcal{J}} P(i, j) \otimes Q(j, k). \tag{1}$$

This "matrix multiplication" is the categorical template behind composition of kernels, Markov kernels, and attention weights.

3.3 Weighted colimits/coends as aggregation

Given an enriched profunctor $A : \mathcal{I}^{\mathrm{op}} \times \mathcal{I} \to \mathcal{W}$ and a feature diagram $H : \mathcal{I} \to \mathcal{V}$, many layers compute

$$H'(i) \;\cong\; \int^{j \in \mathcal{I}} A(i, j) \otimes H(j), \tag{2}$$

followed by pointwise nonlinearities and normalization in $\mathcal{V}$. Equation (2) is a canonical enriched weighted colimit; it specializes to sums, means, max-like constructions (via suitable enrichment), and attention-weighted sums.

4 Neural architectures as open systems: a classification

We now classify common architectures by (i) their interface types and composition regime (open systems viewpoint), and (ii) the structural constraints imposed on their interaction kernels (profunctor viewpoint). These are largely orthogonal design axes.

4.1 A taxonomy at a glance

4.2 Feedforward networks and MLPs

An MLP is typically presented as a composition of affine maps and pointwise nonlinearities:

$$h_{l+1} = \sigma(W_l h_l + b_l).$$

Categorically, this is simply composition of morphisms in a category of (finite-dimensional) vector spaces and smooth maps. In the diagrammatic perspective of [FST19], parametrized functions are morphisms in a symmetric monoidal category; learning procedures become functorial structure on top.

From the open-systems viewpoint, an MLP layer is a component with input port X and output port Y, optionally exposing a parameter port P. Composition of layers is serial wiring.

Architecture	**Index category $\mathcal{I}$**	**Kernel/profunctor constraint**	**Open-systems doctrine**
MLP / feedforward	terminal / trivial	dense linear map (no nontrivial indexing)	deterministic discrete-time
CNN	grid with group action	translation-equivariant (depends on relative offset)	deterministic; often equivariant wiring
GNN	graph category	supported on neighborhoods / adjacency	deterministic; open networked system
Transformer attention	discrete tokens (often with order)	dense, data-dependent weights $A_H(i,j)$	deterministic; global interaction
RNN / LSTM / GRU	time (thin) category	causal (lower-triangular support)	stateful open system; feedback
Neural ODE	time interval	continuous-time flow / integral kernel	continuous-time open system
ResNet	as base model	residual $I + F$ (biproduct/additive structure)	deterministic discrete-time
Mixture-of-experts	expert index + base $\mathcal{I}$	colimit over discrete expert family	deterministic/stochastic gating

Table 1: Survey classification by data shape $\mathcal{I}$, kernel constraints, and doctrine.

4.3 CNNs as equivariant open systems and Kan extensions

Convolutional layers are characterized by (approximate) translation equivariance. In categorical systems terms, equivariance is a constraint on the semantics functor: the action of a group G on the input interface must intertwine with its action on the output interface.

At the kernel level, convolution can be written as a coend indexed by relative offsets:

$$(H')(p) \;\cong\; \int^{\delta} K(\delta) \otimes H(p+\delta),$$

where the kernel K depends only on δ, not on absolute position. This is the same pattern as (2) with $A(p,q) = K(q-p)$. Kan extension language is natural here: a G-equivariant layer is a map between G-objects, and convolution implements a canonical left Kan extension along the action/translation functor (see e.g. equivariance discussions in [FS19]).

4.4 GNNs as neighborhood colimits (message passing)

Message-passing GNN layers have the schematic form

$$h'_v = \phi\Big(h_v, \bigoplus_{u \in \mathcal{N}(v)} m(h_u, h_v)\Big)$$

where $\oplus$ is a permutation-invariant aggregation (sum/mean/max), and $\mathcal{N}(v)$ is a neighborhood.

Let $\mathcal{I}$ be the graph indexing category. For each node v, the neighborhood defines an indexing subcategory $\mathcal{N}(v) \hookrightarrow \mathcal{I}$. Aggregation is a (weighted) colimit of a diagram of messages in $\mathcal{V}$. This realizes the common folklore claim that GNNs are functorial in the graph and that message passing is natural with respect to graph homomorphisms.

From the open-systems viewpoint, a GNN is an *interconnection of many local subsystems* with a fixed wiring pattern (the graph). This matches the "networked open systems" perspective of network models and operadic assembly [Bae+20].

4.5 Transformers: attention as data-dependent profunctor evaluation

Self-attention computes weights

$$A_H(i,j) = \mathrm{softmax}_j\big(\langle Q_H(i), K_H(j)\rangle\big)$$

and outputs

$$H'(i) = \sum_j A_H(i,j)\, V_H(j).$$

Equation (2) matches this exactly when $\mathcal{W} = (\mathbb{R}_{\geq 0}, \cdot)$ (or a suitable enriched setting) and $\otimes$ is scalar multiplication in **Vect**. The novelty relative to CNN/GNN is that the profunctor A_H is *data-dependent*: the kernel is computed from the current representation, not fixed a priori.

From categorical systems theory, Transformers are open systems with global wiring: every token can influence every other token. The constraint distinguishing Transformers from generic kernel methods is largely a *parameterization* of the profunctor A_H (query/key mechanism) and of the post-aggregation pointwise maps (MLPs, layernorm).

4.6 Bi-encoders and cross-encoders as profunctor factorization

Consider a scoring function between two inputs $x \in X$ and $y \in Y$. A cross-encoder computes a score using joint interaction, and in categorical terms can be regarded as an arbitrary enriched profunctor

$$S : X^{\mathrm{op}} \times Y \to \mathcal{W}.$$

A bi-encoder restricts to profunctors that factor through a representation functor:

$$S(x,y) = \langle E(x), F(y)\rangle$$

for encoders $E : X \to \mathbf{Vect}$, $F : Y \to \mathbf{Vect}$. This is a representability/low-rank constraint: the general profunctor S is replaced by one induced from a pairing in **Vect**. The inclusion "bi-encoder kernels $\subsetneq$ cross-encoder kernels" becomes a statement about the expressive class of profunctors allowed.

4.7 Autoencoders and variational autoencoders

An autoencoder factors an endomorphism through a latent space:

$$X \xrightarrow{e} Z \xrightarrow{d} X, \qquad d \circ e \approx \mathrm{id}_X.$$

In categorical systems terms, this is an *open system with a hidden interface* Z, together with an approximate section/retraction property. It is more accurate to regard "autoencoder" as a *training template* (a constraint on behavior), rather than as a base architecture in the sense of Table 1.

A VAE replaces e, d with stochastic channels $q(z \mid x)$, $p(x \mid z)$. This is naturally expressed in a Markov category [Fri20] (or related CD-categories [CJ19]), where composition of channels is the monoidal categorical composition and marginalization/integration appears as categorical structure (often via coends or string diagrams).

4.8 Diffusion and Markovian generative models

Diffusion models and related score-based methods can be viewed (at a coarse categorical level) as learning a family of reverse-time Markov kernels $p_\theta(x_{t-1} \mid x_t)$ whose composition over time yields a generative process. Markov categories provide a compositional language for such kernels [Fri20], and the string-diagrammatic perspective on Bayesian inversion/disintegration [CJ19] is directly aligned with learning conditional distributions.

4.9 Residual connections and additive structure

Residual layers $h' = h + F(h)$ rely on additive/biproduct-like structure in the feature category. At the kernel level, this corresponds to adding the identity profunctor to a learned one. In systems terms, this is parallel composition followed by a merge/combine map, reminiscent of the Frobenius structure present in hypergraph categories [Fon15; Fon18].

4.10 Recurrent networks, feedback, and traced structure

RNNs expose an explicit state that persists across time. Categorical models of feedback typically use traced monoidal categories or related double categorical structure; Myers' double category approach to open dynamical systems is designed to support such constructions [Mye21]. In the "index category" lens, RNNs correspond to causal kernels supported on $j \leq i$ in a thin time category.

4.11 Mixtures-of-experts as colimits over an expert index

Mixture-of-experts models compute outputs as a weighted combination of expert functions:

$$f(x) = \sum_{k \in K} g_k(x)\, f_k(x).$$

Categorically, this is a weighted colimit over a discrete index category K, with a gating mechanism producing the weights. This clarifies that MoE is not a competing "base architecture" to Transformers or CNNs; it is a *colimit-based assembly pattern* that can be applied on top of a base interaction kernel.

5 Open systems, black-boxing, and compositional semantics

A recurring theme in categorical systems theory is the separation between a *syntactic* category of open systems (wiring diagrams, decorated cospans) and a *semantic* category of behaviors (relations, linear relations, stochastic relations), connected by a black-boxing functor. Baez–Pollard's reaction network framework includes black-boxing to steady-state relations [BP17]; analogous black-boxing appears for passive linear networks [BF18] and open Markov processes [BFP16].

For neural networks, one may regard:

- the *syntactic* level as the computation graph with explicit ports (inputs/outputs, possibly parameters), and
- the *semantic* level as the induced function f_θ (or channel) between external interfaces.

The categorical formalism clarifies that "architecture" lives primarily at the syntactic level (interface types + wiring + constraints), while "objective" constrains semantic behavior.

6 Process theories and diagrammatic reasoning

Coecke's "process theories" popularize string-diagrammatic reasoning for compositional systems [CK17]. In NLP, the DisCoCat program casts compositional semantics as morphisms in compact closed categories, enabling diagrammatic accounts of information flow [CSC10]. These ideas connect naturally to neural representation learning, especially when attention and similarity are expressed as pairings and coends.

From the perspective of this survey, the principal takeaway is methodological: the same graphical calculus that supports reasoning about circuits, stochastic processes, and quantum protocols can also express neural architectures once they are presented as open systems in a suitable monoidal category.

7 Discussion: clearing up "architecture" in ML

The categorical systems view suggests a clean decomposition of what ML practice often lumps together:

1. **Base interaction law** (kernel/profunctor constraint): local vs global, equivariant vs generic, sparse vs dense.
2. **System doctrine** (dynamics): feedforward discrete-time, recurrent with feedback, continuous-time, stochastic.
3. **Assembly patterns**: residual addition, pooling as colimits, mixtures as weighted colimits, multi-input scoring as profunctors.
4. **Training templates**: autoencoding, contrastive training, diffusion objectives—all constraints on semantics rather than on syntax.

This separation helps interpret common phrases like "Transformer bi-encoder" or "graph diffusion model": they combine choices along multiple axes rather than naming a single monolithic architecture.

Limitations. While the coend/profunctor viewpoint unifies many aggregation-based layers, real neural systems include normalization, gating, and nonlinearities that are best modeled as additional morphisms or as structure in the feature category $\mathcal{V}$. Moreover, data-dependent kernels (attention, routing, adaptive computation) point toward higher categorical structure (e.g. fibrations or indexed profunctors) beyond the scope of this survey.

Outlook. Systems doctrine frameworks [Mye23] suggest a principled way to compare deterministic and stochastic models, discrete- and continuous-time models, and hybrid constructions under one umbrella. For ML, a mature categorical survey could further integrate differentiable programming, optimization-as-dynamics, and semantics of training updates as in [FST19].

References

[Bae+20] J. C. Baez, J. Foley, J. Moeller, and B. S. Pollard. "Network Models". In: *Theory and Applications of Categories* 35.20 (2020), pp. 700–744. arXiv: 1711.00037 [math.CT]. URL: https://arxiv.org/abs/1711.00037.

[BC20] J. C. Baez and K. Courser. "Structured Cospans". In: *Theory and Applications of Categories* 35.48 (2020), pp. 1771–1822. DOI: 10.70930/tac/dca3obx4. arXiv: 1911.04630 [math.CT].

[BCV22] J. C. Baez, K. Courser, and C. Vasilakopoulou. "Structured versus Decorated Cospans". In: *Compositionality* 4.3 (2022). DOI: 10.32408/compositionality-4-3. arXiv: 2101.09363 [math.CT].

[BF18] J. C. Baez and B. Fong. "A Compositional Framework for Passive Linear Networks". In: *Theory and Applications of Categories* 33.38 (2018), pp. 1158–1222. arXiv: 1504.05625 [math.CT]. URL: https://arxiv.org/abs/1504.05625.

[BFP16] J. C. Baez, B. Fong, and B. S. Pollard. "A Compositional Framework for Markov Processes". In: *Journal of Mathematical Physics* 57.3 (2016), p. 033301. DOI: 10.1063/1.4941578. arXiv: 1508.06448 [math-ph].

[BP17] J. C. Baez and B. S. Pollard. "A Compositional Framework for Reaction Networks". In: *Reviews in Mathematical Physics* 29.9 (2017), p. 1750028. DOI: 10.1142/S0129055X17500283. arXiv: 1704.02051 [math-ph].

[CJ19] K. Cho and B. Jacobs. "Disintegration and Bayesian inversion via string diagrams". In: *Mathematical Structures in Computer Science* 29.7 (2019), pp. 938–971. DOI: 10.1017/S0960129518000488. arXiv: 1709.00322 [math.CT].

[CK17] B. Coecke and A. Kissinger. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017. ISBN: 9781107104228.

[CSC10] B. Coecke, M. Sadrzadeh, and S. Clark. "Mathematical Foundations for a Compositional Distributional Model of Meaning". In: *Linguistic Analysis* 36 (2010), pp. 345–384. arXiv: 1003.4394 [cs.CL].

[Fon15] B. Fong. "Decorated Cospans". In: *Theory and Applications of Categories* 30.33 (2015), pp. 1096–1120. arXiv: 1502.00872 [math.CT]. URL: https://arxiv.org/abs/1502.00872.

[Fon18] B. Fong. "Decorated Corelations". In: *Theory and Applications of Categories* 33.22 (2018), pp. 608–643. arXiv: 1703.09888 [math.CT]. URL: https://arxiv.org/abs/1703.09888.

[Fri20] T. Fritz. "A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics". In: *Advances in Mathematics* 370 (2020), p. 107239. DOI: 10.1016/j.aim.2020.107239. arXiv: 1908.07021 [math.ST].

[FS19] B. Fong and D. I. Spivak. *Seven Sketches in Compositionality: An Invitation to Applied Category Theory*. Cambridge University Press, 2019. ISBN: 9781108711821. DOI: 10.1017/9781108668804. arXiv: 1803.05316 [math.CT].

[FST19] B. Fong, D. I. Spivak, and R. Tuyéras. "Backprop as Functor: A compositional perspective on supervised learning". In: *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*. 2019, pp. 1–13. DOI: 10.1109/LICS.2019.8785665. arXiv: 1711.10455 [math.CT].

[Kel82] G. M. Kelly. *Basic Concepts of Enriched Category Theory*. Repr. Theory Appl. Categ. No. 10 (2005). Cambridge University Press, 1982.

[Mye21] D. J. Myers. "Double Categories of Open Dynamical Systems (Extended Abstract)". In: *Proceedings of Applied Category Theory 2020*. Vol. 333. Electronic Proceedings in Theoretical Computer Science. 2021, pp. 154–167. DOI: 10.4204/EPTCS.333.11. arXiv: 2005.05956 [math.CT].

[Mye23] D. J. Myers. *Categorical Systems Theory (draft book)*. https://www.davidjaz.com/Papers/DynamicalBook.pdf. Draft manuscript. 2023.

[RS13] D. Rupel and D. I. Spivak. "The operad of temporal wiring diagrams: formalizing a graphical language for discrete-time processes". In: *arXiv preprint* (2013). arXiv: 1307.6894 [math.CT]. URL: https://arxiv.org/abs/1307.6894.

[Spi13] D. I. Spivak. "The operad of wiring diagrams: formalizing a graphical language for databases, recursion, and plug-and-play circuits". In: *arXiv preprint* (2013). arXiv: 1305.0297 [cs.DB]. URL: https://arxiv.org/abs/1305.0297.

[VSL15] D. Vagner, D. I. Spivak, and E. Lerman. "Algebras of Open Dynamical Systems on the Operad of Wiring Diagrams". In: *Theory and Applications of Categories* 30.51 (2015), pp. 1793–1822. arXiv: 1408.1598 [math.CT]. URL: https://arxiv.org/abs/1408.1598.