Proceedings Track

Compositional Symmetry as Compression: Lie-Pseudogroup Structure in Algorithmic Agents

Abstract

In the algorithmic (Kolmogorov) view, agents are programs that track and compress sensory streams using generative programs. We propose a framework where the relevant structural prior is simplicity (Solomonoff) as *compositional symmetry*, where natural streams are well described by (local) actions of finite-parameter Lie pseudogroups on geometrically and topologically complex low-dimensional configuration manifolds (latent spaces). Modeling the agent as a generic neural dynamical system coupled to such streams, we show that accurate world-tracking imposes (i) *structural* constraints (equivariance of the agent system constitutive equations and readouts) and (ii) *dynamical* constraints: under static inputs, symmetry induces conserved quantities (Noether-style labels) in agent dynamics and confines trajectories to reduced invariant manifolds; under slow drift, these manifolds move but remain low-dimensional. This yields a hierarchy of reduced manifolds aligned with the compositional factorization of the pseudogroup—a geometric account of the "blessing of compositionality" in deep models. We connect these ideas, at a high level, to the Spencer formalism for Lie pseudogroups, and formulate a symmetry-based, self-contained version of predictive coding in which higher layers receive only *coarse-grained residual transformations* (prediction-error coordinates) along symmetry directions unresolved at lower layers.

Keywords: Kolmogorov Theory of Consciousness, Lie pseudogroups, Compositional symmetry, Kolmogorov complexity, Hierarchical reduction, manifold hypothesis, coarse-graining, predictive coding, compression

1. Introduction

Kolmogorov Theory (KT) proposes a scientific framework centered on algorithmic agents using the mathematics of algorithmic information theory (AIT), where the central concepts are computation and compression. In particular, the Kolmogorov complexity of a data object— the length of the shortest program producing the data object (Cover and Thomas, 2006)—provides a powerful conceptual anchor in the theory. In KT, an algorithmic agent is an information-processing system that builds and runs *compressive* models (programs) of its environment, using them to guide action and shape experience (see Figure 1). Model building serves an Objective Function (telehomeostasis, or preservation of pattern, in living systems) and is guided by Ockham's Razor (simpler models/shorter programs are preferred) (Sober, 2015; **?**; **?**).

A *model* is the *short program* that generates a given dataset. The same program can be read as a smooth generative map $f : \mathcal{C} \to \mathbb{R}^X$, turning parameters into data object instances. A third view is symmetry-based: the generator induces a group of automorphisms whose orbits cover the data manifold; learning invariances recovers the generator (**?**). When no short
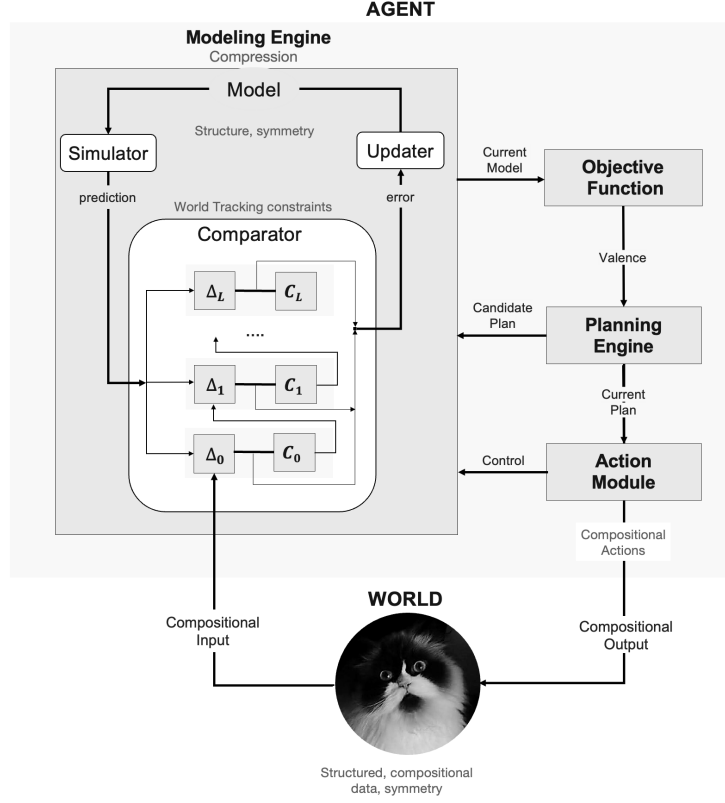
Figure 1: The **algorithmic agent** interacts with the World (structure, symmetry, compositional data) (**???**). The *Modeling Engine* (compression) runs the current Model (which encodes found structure/symmetry), makes hierarchical/compositional predictions of future coarse-grained (pooled) data, and then evaluates the prediction error in the *Comparator* (world-tracking constraint monitoring) to hierarchically update the Model. The comparison process is carried *hierarchically*, and the output is coarse-grained to feed the next level. We reflect this process mathematically as a world-tracking constraint on the dynamics (Equation (3) **Hierarchical modeling engine (Comparator).** Level k predicts $\hat{I}_k = \hat{\gamma}_k \cdot I_0$, compares to its incoming datum (raw image at $k = 0$; canonicalized, coarse-grained residual $m_{k-1 \to k}$ for $k \geq 1$), updates from error E_k, and forwards only the residual after canonicalization and coarse-graining, $m_{k \to k+1} = \mathcal{C}_{k \to k+1}(\hat{\gamma}_k^{-1} \cdot \text{input} - I_0)$. Generators shrink along $H_0 \supset H_1 \supset H_2 \cdots$; residuals live in quotient directions and carry "what's left to explain" to coarser scales. The *Planning Engine* runs counterfactual simulations and selects plans for the next (compositional) actions (agent outputs to world and self). The Updater receives hierarchical prediction errors from the Comparator as inputs to improve the Model. All modules can be implemented hierarchically.

deterministic program exists, the optimal compressor is statistical; expected Kolmogorov complexity converges to entropy rate (Cover and Thomas, 2006; Li and Vitanyi, 2007).

Proceedings Track

A central question is: *what structure can such compressive models exploit?* The language of group theory provides a good handle to formalize the notion of structure. Discrete groups admit *Cayley graphs* encoding generators and relations (Biggs, 1974). For Lie groups, infinitesimal structure lies in a finite-dimensional Lie algebra; for semisimple algebras, *Dynkin diagrams* encode simple roots and relations (Humphreys, 1972; Knapp, 2002). Lie *pseudogroups*—local transformation groups defined by involutive PDEs—admit infinitesimal descriptions that can be organized in the *Spencer complex*, whose differentials encode compatibility conditions (Goldschmidt, 1967; Seiler, 2010).

Natural streams exhibit *compositional symmetry*: pose, viewpoint, articulations, and semantic deformations compose recursively and act locally (Simon, 1991; Riesenhuber and Poggio, 1999; Poggio et al., 2020; Cagnetta et al., 2024; **?**). This points beyond global (finite) Lie groups to *Lie pseudogroups* of local diffeomorphisms as a language to describe structure, which can be captured (when needed) in the Spencer framework. This framework provides a jet-level *compatibility complex* (the Spencer complex)—a quiver-like graded sequence of bundles and differential operators whose cohomology governs formal integrability, prolongation, and obstructions—organising how local symmetries compose and constrain dynamics across scales. In plainer terms, it acts as a structured *consistency checker*: it tracks, order by order in derivatives, whether local symmetry rules can be stitched into a coherent global transformation, and pinpoints exactly where that stitching fails.

We take the persepective of Lie pseudogroups as a programming language to describe generative models and to motivate a built-in modeling hierarchy. Here we propose a symmetry-first account of model structure for algorithmic agents where we (i) define *generative models* as local actions of finite-parameter Lie pseudogroups on configuration manifolds, (ii) model the agent as a neural ODE driven by such streams with a *Comparator* enforcing tracking, and (iii) derive the *constitutive* (equivariance) and *dynamical* (Noether-style invariants, reduced manifolds) constraints that follow. This structure explains why deep, hierarchical architectures—which mirror compositional symmetry—attain favorable sample complexity on hierarchical tasks (Poggio et al., 2016; Poggio and Fraser, 2024), and why a *bare* manifold prior can be insufficient without additional geometric covering structure (Kiani et al., 2024). The analysis suggests symmetry-aware designs (e.g., group-equivariant networks) as principled architectures for agents and offers a group-theoretical lens on predictive coding. Finally, we provide a tentative implementation of hierarchical predictive processing Friston (2018) using this formalism. For an overview of the logical structure of the paper, see Figure 2.

Intuition. Compositional symmetry is a *program schema*: complex transformations are built by nesting a few primitive moves near the identity. Agents that align their internal structure with this recursive grammar compress better, learn with fewer samples, and generalize over orbits of the same symmetry.

2. Generative Models as Lie-Pseudogroup Actions

Natural data will appear low-dimensional (the manifold hypothesis) if generated by *hierarchical continuous symmetries* generated by a finite dimensional Lie pseudogroup. We

capture this through the local actions of a finite-parameter Lie pseudogroup G on a configuration manifold $\mathcal{C}$:

Definition 1 (Generative model) *A generative model is a smooth map $f : \mathcal{C} \rightarrow \mathbb{R}^X$ from an M-dimensional configuration manifold ($M \ll X$) to observations. For $c \in \mathcal{C}$, write $I = f(c) \in \mathbb{R}^X$.*

Let G act locally on $\mathcal{C}$ and on $\mathbb{R}^X$. Pick $c_0 \in \mathcal{C}$ with $I_0 = f(c_0)$.

Definition 2 (Lie generative model) *The map f (or $f(\mathcal{C})$) is a Lie generative model if*

$$\forall c \in \mathcal{C} \; \exists \gamma \in G : \quad c = \gamma \cdot c_0, \quad f(c) = \gamma \cdot I_0.$$

Equivalently, $I = f(\gamma \cdot c_0) = \gamma \cdot I_0$ with $\gamma \in G$.

Intuition (recursion and compression). Because G is a Lie (pseudo)group, elements near the identity factor as

$$\gamma = \exp\Big(\sum_{k=1}^{r} \theta_k T^k\Big),$$

so complex deformations arise by composing *infinitesimal* ones (Hall, 2015). Here $\gamma(\cdot) : U \rightarrow G$ is a local parameterization of G, and "$\cdot$" denotes the induced (generally nonlinear) action of G on image space. This endows f with a recursive, compositional parameterization: a short, structured program in the sense of algorithmic information theory. Pseudogroups permit this *locally* even when global topology precludes a single global action (Seiler, 2010).

If natural data is generated by a "random walk" through the parameter space of a Lie pseudogroup, the following apply: (i) *Compression:* the data can be compressed, because r parameters span the data manifold. (ii) *Efficient learning:* deep architectures that respect generators can enjoy the *blessing of compositionality* (Poggio et al., 2016; Poggio and Fraser, 2024; Miao and Rao, 2007; Anselmi and Poggio, 2022)—fewer data (sample complexity) is needed for learning with the right architecture). (iii) *Symmetry discovery:* when G is unknown, its generators can be potentially be inferred (Moskalev et al., 2022). We study the dynamical implications for the world-tracking agent in the next section.

3. Symmetry-Constrained World-Tracking Dynamics

Here, we assume natural input/sensory streams are not arbitrary but arise from finite-parameter Lie (pseudo)group generators acting on a low-dimensional configuration manifold (Sec. 2) and show that an algorithmic agent that can *track* such streams need to be compatible with their underlying symmetry. Let the external data stream be generated by a finite-parameter Lie pseudogroup G acting on a reference image I_0,

$$I_{\theta(t)} = \gamma(\theta(t)) \cdot I_0, \qquad \theta(t) \in U \subset \mathbb{R}^r \text{ (local chart on } G\text{)}. \tag{1}$$

The agent's high-dimensional state $x \in \mathbb{R}^X$ follows the general neural network equation

$$\dot{x} = F\big(x; \, w, \, I_{\theta(t)}\big), \tag{2}$$
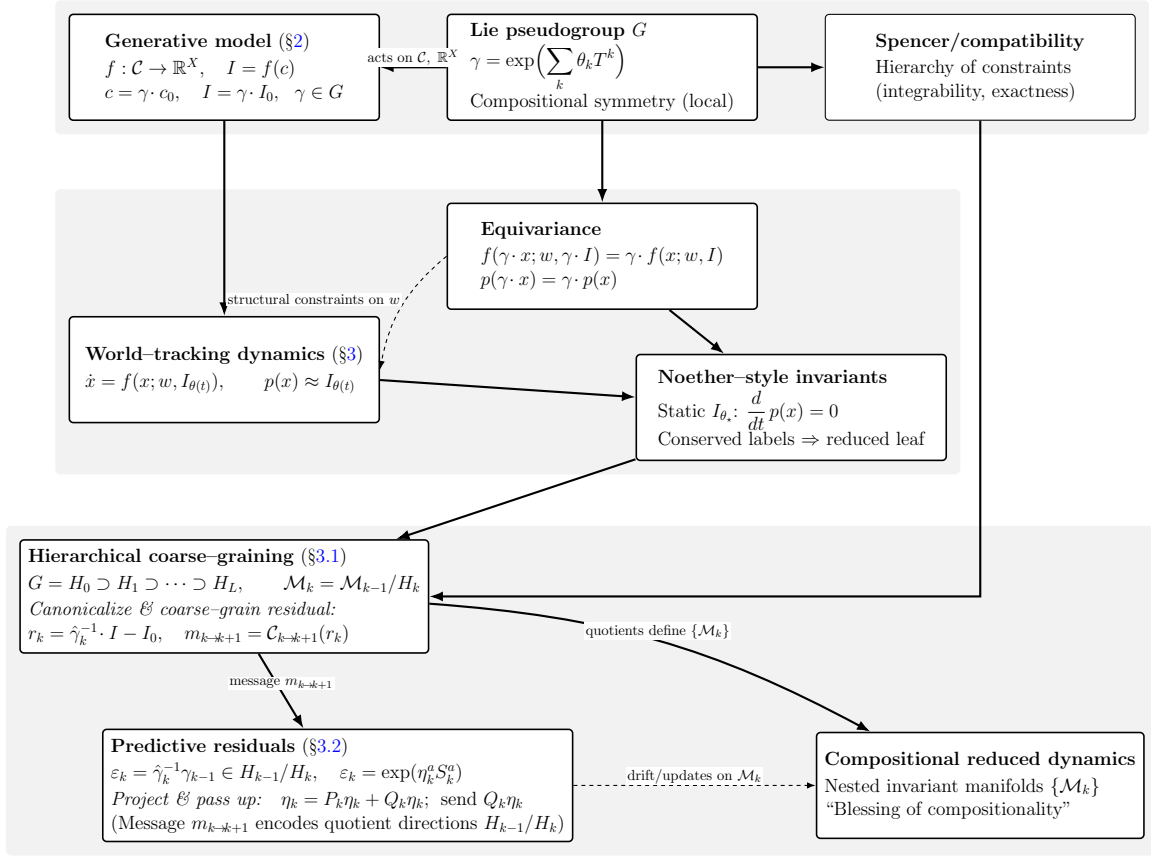
4

Proceedings Track

Figure 2: **From generative symmetry to compositional reduced dynamics.** Top: a Lie pseudogroup G acts locally on configuration $\mathcal{C}$ and observation space, with Spencer providing a hierarchy of compatibility constraints. Middle: the agent enforces world tracking (Eq. (3)); equivariance (Eq. (4)) imposes structural constraints and, for static inputs, yields conserved labels (Noether-style), confining trajectories to reduced leaves. Bottom: a flag $G = H_0 \supset \cdots \supset H_L$ *defines* the nested invariant manifolds $\{\mathcal{M}_k\}$ (solid bent arrow); residual messages $m_{k \to k+1}$ parameterize quotient directions $\varepsilon_k \in H_{k-1}/H_k$ and *induce drift/updates* on these leaves (dashed arrow), yielding a hierarchy of compositional reduced dynamical manifolds.

with fixed weights w. A projector $p : \mathbb{R}^X \to \mathbb{R}^Y$ $(Y \ll X)$ provides the *world-tracking equation*

$$p\big(x(t)\big) \ \approx \ I_{\theta(t)}. \tag{3}$$

This equation states that the agent is able to lock into the input data stream.

Compatibility of tracking with invariants requires equivariance. Equivariance of Equation 2 says "solutions come in G–families." Uniqueness lets us *name* each solution by the group element needed to reach it from a reference (for simplicity we consider connected groups). These names are constant along trajectories (conserved).

The tracking constraint (Equation 3) is feasible precisely when it latches onto these constants; otherwise it fights the flow. Under static inputs the constraint $p(x) = I_{\theta_\star}$ can be sustained only if it *depends on* the conserved labels (cyclic coordinates) induced by the symmetry of the equations. Otherwise it would over-constrain trajectories to a point. In canonical coordinates adapted to the symmetry, $p(x)$ must be a function of the corresponding constants of motion. With slowly varying $\theta(t)$ the labels become *adiabatic invariants*, and the constrained leaf *drifts* but remains low-dimensional.

Consequences of equivariance requirement. Because I_θ is moved by G (the equation must hold for any γ), effective tracking demands that the internal dynamics respect the same action,

$$\forall \gamma \in G: \ f(\gamma \cdot x; \ w, \gamma \cdot I_\theta) = \gamma \cdot f(x; \ w, I_\theta), \quad p(\gamma \cdot x) = \gamma \cdot p(x). \tag{4}$$

Eq. (4) places *structural constraints* on w (weight tying, zero blocks, etc.) identical in spirit to group-equivariant CNNs (Moskalev et al., 2022; **?**). For infinitesimal $\gamma = \exp(\epsilon T_k)$ we obtain linear commutation conditions $T_k f = (\partial f / \partial x) T_k x$ that must hold for every Lie-algebra generator T_k.

Conservation laws. Freeze θ so that $\gamma(t) = \gamma_0$. Under Eq. (4) the read-out becomes invariant: $p(x) = \text{const}$. Hence, each of the Y read-out channels defines a conserved quantity. This is a direct Noether analogue: continuous symmetry $\gamma \Rightarrow$ invariants $\{p_j(x)\}_{j=1}^{Y}$ (Hydon, 2000). Trajectories are confined to an $(X-Y)$-dimensional leaf of phase space. When $\theta(t)$ varies slowly, the invariant leaf drifts, but its dimension remains $\leq M + X - Y$ (gains at most M extra degrees of freedom).

Approximate tracking via Lyapunov control. To formalize the notion of approximate tracking after transient dynamics ($p(x) \approx I_\theta$), we can use the machinery of Lyapunov functions. We modify the constrained dynamical equations above by defining error and Lyapunov functions and add a symmetry-preserving feedback term

$$E = p(x) - I_\theta, \quad V = \tfrac{1}{2}\|E\|^2, \quad \dot{x} = F(x; w, I_\theta) + K E, \tag{5}$$

with a gain operator K that commutes with every T_k. We then need to choose a gain K such that $D_x p(x) K$ is negative semi-definite and K commutes with the induced action of each generator T_k (so the tracking equations remain equivariant). Then $\dot{V} \leq 0$ under static inputs; with slow drift one obtains an ISS-type bound $\dot{V} \leq -\alpha\|E\|^2 + \beta\|\dot{I}_\theta\|^2$ for suitable $\alpha, \beta > 0$ (Sontag, 2008).

Remark. All constructions are local: actions, exponentials, and invariant leaves are taken in the chart domains where the pseudogroup is defined; global statements require additional compatibility (Spencer exactness).

Summary. Equivariance (4) forces **structural** constraints on the agent (weights obey group commutation) and **dynamical** constraints (conserved read-outs and low-dimensional invariant manifolds). A Lyapunov closure (5) formalises the "$\approx$" in (3), completing a compact, symmetry-first description of world-tracking dynamics.

3.1. Lie-pseudogroups, hierarchical coarse-graining, and reduced manifolds

Compositionality enters via generator nesting: every $\gamma \in G$ near the identity factors as $\exp(\sum_k \theta_k T^k)$. A *hierarchical description* is obtained by declaring certain generators neg-

Proceedings Track

ligible at coarser levels. Formally, choose a flag of sub-pseudogroups

$$G = H_0 \supset H_1 \supset \cdots \supset H_L, \tag{6}$$

where H_k is obtained from H_{k-1} by omitting some generators. Let $\mathcal{M}_0$ denote the full state space; define the reduced manifold at level k as the quotient $\mathcal{M}_k := \mathcal{M}_{k-1}/H_k$ (locally: flows of the discarded generators are frozen). Then

$$\mathcal{M}_0 \supset \mathcal{M}_1 \supset \cdots \supset \mathcal{M}_L, \qquad \dim \mathcal{M}_{k+1} < \dim \mathcal{M}_k, \tag{7}$$

and motion within $\mathcal{M}_k$ is driven only by residual generators in H_k. In the formal theory of differential constraints, the flag $H_0 \supset \cdots \supset H_L$ is encoded by the Spencer sequence of G; exactness of that sequence guarantees compatibility of the layered constraints and integrability of the reduced dynamics. Hence

Lie pseudogroup $\implies$ Spencer tower of sub-pseudogroups $\implies$ nested invariant manifolds

Each quotient sharpens the agent's description while retaining the ability to *recompose* fine structure by re-activating generators, capturing the "blessing of compositionality": models learn efficiently on $\mathcal{M}_k$ yet can recover detail by ascending the tower. In this sense, hierarchical cognition, coarse–grained abstraction, and the algebraic–geometric machinery of Lie pseudogroups are three facets of the same structure.

Intuition. Each level "throws out" a set of generators, fixes the associated conserved labels, and descends to a simpler manifold. The tower mirrors the grammar of G: coarse levels use fewer letters; fine levels re-introduce them as needed. Spencer's framework encodes such hierarchical compatibility; exactness corresponds to the absence of obstructions in the layered constraints (Seiler, 2010).

3.2. Predictive hierarchy as residual transformations

Here, we provide a tentative formalization of the predictive processing hierarchy. As before, all statements are local in the domain where the pseudogroup action and charts overlap; residuals and compositions are taken in that neighborhood.

Incoming data are processed bottom-up (see Figure 1). The most detailed layer predicts fine features using its own generators, compares prediction to input, and forms an *error*: the part it could not explain. That error is used locally to refine the layer's hypothesis; only the *unexplained remainder* is coarse-grained (pooled) and passed upward. The next layer absorbs what it can with its own generators and again forwards just the coarse-grained residual:

predict $\rightarrow$ compare $\rightarrow$ update locally $\rightarrow$ canonicalize residual $\rightarrow$ coarse-grain $\rightarrow$ pass up

In steady scenes the residuals shrink toward zero at all levels, leaving stable labels (conserved coordinates); a persistent residual at the top flags a missing generator or a model mismatch.

7

Set–up and linear residual fit. Let $G = H_0 \supset H_1 \supset \cdots \supset H_L$ be a nested family of (local) sub-pseudogroups with Lie algebras $\mathfrak{g} \supset \mathfrak{h}_1 \supset \cdots \supset \mathfrak{h}_L$. At time t the datum is $I_{\theta(t)} = \gamma_{\text{true}}(t) \cdot I_0$, while level k predicts $\hat{I}_k(t) = \hat{\gamma}_k(t) \cdot I_0$ with $\hat{\gamma}_k(t) \in H_k$. The observation–space error is $\delta I_k = I_{\theta(t)} - \hat{I}_k(t)$. Because the stream is (locally) generated by a group action, small errors are explained by a small *residual* $\varepsilon_k = \exp(\eta_k)$ near the identity, so that $I_{\theta(t)} \approx (\hat{\gamma}_k \, \varepsilon_k) \cdot I_0$. Linearizing the action at $\hat{I}_k$ with a basis $\{S_a\}$ of $\mathfrak{g}$ and induced image–space velocities $V_a(\hat{I}_k)$, $\delta I_k \approx \sum_a \eta_k^a \, V_a(\hat{I}_k)$, so we obtain $\eta_k = (\eta_k^a)$ by a regularized least–squares fit (e.g., Tikhonov regularization).

Within–level update via a symmetry–respecting projection. Level k can realize only directions in $\mathfrak{h}_k = \text{Lie}(H_k)$. Fix an inner product on $\mathfrak{g}$ that is $\text{Ad}(H_k)$–invariant and let $P_k : \mathfrak{g} \to \mathfrak{h}_k$ be the associated orthogonal projector (with $Q_k := I - P_k$). The realizable part updates the hypothesis, $\hat{\gamma}_k \leftarrow \hat{\gamma}_k \exp(P_k \eta_k)$. The *unresolved* component $Q_k \eta_k$ is what must be communicated upward.

Canonicalize and coarse–grain before passing up. To present a clean message at the next scale, first express the residual in a common reference frame (canonicalization),

$$r_k := \hat{\gamma}_k^{-1} \cdot I_{\theta(t)} - I_0, \tag{8}$$

which removes the frame explained at level k. Then apply an H_k–invariant, H_{k+1}–equivariant coarse–grainer $\mathcal{C}_{k \to k+1} : \mathbb{R}^Y \to \mathbb{R}^{Y_{k+1}}$ and define the upward message

$$m_{k \to k+1} := \mathcal{C}_{k \to k+1}(r_k). \tag{9}$$

Intuitively, $\mathcal{C}_{k \to k+1}$ "irons out" fine variation along H_k–orbits while preserving precisely the structure modelled at level $k+1$. The next layer treats $m_{k \to k+1}$ as its comparator datum and repeats the same loop with its generators. Equivalently, one may pass an algebra–space message $\bar{\eta}_{k \to k+1} = \text{Ad}_{\hat{\gamma}_k^{-1}}(Q_k \eta_k)$; we use the observation–space version for numerical stability and alignment with predictive coding practice.

4. Discussion

Starting from the premise that sensory streams are generated by finite-parameter Lie (pseudo)group actions, we offer a symmetry-aware, group-theoretic account of compression and tracking, with the additional benefit of mathematica tools to describe structure in models. Under static inputs, equivariance of the closed-loop vector field yields Noether-style conserved labels (solution-symmetry invariants), and the tracking constraint is feasible precisely when it depends on those labels; trajectories lie on reduced invariant manifolds. Under slow drift, the labels become adiabatic, and the manifold drifts with at most M added degrees of freedom. Structural equivariance constrains parameters, guiding architecture design (e.g., equivariant layers). The hierarchical quotient $\mathcal{M}_0 \supset \cdots \supset \mathcal{M}_L$ formalizes compositionality and explains the favorable sample complexity of deep, symmetry-respecting models (Poggio et al., 2016; Poggio and Fraser, 2024), while clarifying why a manifold prior alone can be insufficient (Kiani et al., 2024). We provide a fuller description of the formalization of hierarchy in Appendix A, as well as a conceptual example in Appendix C. We provided a tentative implementation of predictive coding using this formalism, where

the hierarchical structure of the generative model is used "backwards" (see Appendix 3.1). Finally, although we did not discuss this here, methods for symmetry discovery (Moskalev et al., 2022; Hu et al., 2024) provide a route to *learn G* from data, closing the loop between structure and learning.

5. Conclusions and Future Directions

By defining generative models using group theory, we linked compositional symmetry, Lie pseudogroups, and hierarchical reduction to a precise dynamical picture of world-tracking. Although this is certainly possible in some cases (e.g., tracking robotic, jointed cats), it may fail when the generative model's latent space is very complex. Future work may include: (i) generalization to stochastic inputs and analysis of robustness (SDE analogues of (5)); (ii) development of K operators for valid Lyapunov world tracking problems; (iii) empirical tests with equivariant architectures under controlled generative symmetries; (iv) formal links to Spencer exactness, moduli stacks and integrability guarantees in practical learning systems.

Acknowledgments

References

Fabio Anselmi and Tomaso Poggio. Representation learning in sensory cortex: A theory. *IEEE access : practical innovations, open solutions*, 10:102475–102491, 2022. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

Norman L. Biggs. *Algebraic graph theory*. Cambridge University Press, Cambridge, 1974. ISBN 0-521-20335-X.

Francesco Cagnetta, Leonardo Petrini, Umberto M. Tomasini, Alessandro Favero, and Matthieu Wyart. How Deep Neural Networks Learn Compositional Data: The Random Hierarchy Model. *Physical Review X*, 14(3):031001, July 2024. doi: 10.1103/PhysRevX. 14.031001. URL https://link.aps.org/doi/10.1103/PhysRevX.14.031001. Publisher: American Physical Society.

Blender Online Community. Blender - a 3D modelling and rendering package. manual, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL http://www.blender.org.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & sons, 2 edition, 2006. tex.date-added: 2016-10-20 21:00:29 +0000 tex.date-modified: 2016-10-20 21:00:58 +0000.

Karl Friston. Does predictive coding have a future? *Nature Neuroscience*, 21(8):1019–1021, August 2018. ISSN 1097-6256, 1546-1726. doi: 10.1038/s41593-018-0200-7. URL https://www.nature.com/articles/s41593-018-0200-7.

Hubert Goldschmidt. Integrability criteria for systems of nonlinear partial differential equations. *Journal of Differential Geometry*, 1(3):269–307, 1967. doi: 10.4310/jdg/1214428094.

Brian C. Hall. *Lie groups, lie algebras, and representations: An elementary introduction.* Springer, 2015.

Lexiang Hu, Yikang Li, and Zhouchen Lin. Symmetry Discovery for Different Data Types, October 2024. URL http://arxiv.org/abs/2410.09841. arXiv:2410.09841 [cs].

James E. Humphreys. *Introduction to lie algebras and representation theory*, volume 9 of *Graduate texts in mathematics*. Springer-Verlag, New York, 1972. ISBN 3-540-90053-5.

Peter E Hydon. *Cambridge texts in applied mathematics: Symmetry methods for differential equations: A beginner's guide series number 22.* Cambridge texts in applied mathematics. Cambridge University Press, Cambridge, England, January 2000.

Bobak T. Kiani, Jason Wang, and Melanie Weber. Hardness of learning neural networks under the manifold hypothesis, 2024. URL https://arxiv.org/abs/2406.01461. arXiv: 2406.01461 [cs.LG].

Anthony W. Knapp. *Lie groups beyond an introduction*, volume 140 of *Progress in mathematics*. Birkhäuser, Boston, 2 edition, 2002. ISBN 0-8176-4259-5.

Ming Li and Paul M.B. Vitanyi. Applications of algorithmic information theory. *Scholarpedia*, 2(5):2658, 2007. tex.date-added: 2016-10-15 14:12:39 +0000 tex.date-modified: 2016-10-22 13:13:18 +0000.

Kevin M Lynch and Frank C Park. *Modern robotics.* Cambridge University Press, Cambridge, England, May 2017.

Xu Miao and Rajesh P N Rao. Learning the Lie groups of visual invariance. *Neural Computation*, 19(10):2665–2693, October 2007. Publisher: MIT Press - Journals.

Artem Moskalev, Anna Sepliarskaia, Ivan Sosnovik, and Arnold Smeulders. LieGG: Studying learned Lie group generators. *arXiv*, 2022. Publisher: arXiv.

T Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep – but not shallow –Networks avoid the curse of dimensionality: a review. *CBMM Memo*, (058), 2016. tex.date-added: 2016-12-04 00:19:36 +0000 tex.date-modified: 2016-12-04 00:21:03 +0000.

Tomaso Poggio and Maia Fraser. Compositional sparsity of learnable functions. *Bulletin of the American Mathematical Society*, 61(3):438–456, July 2024. ISSN 0273-0979, 1088-9485. doi: 10.1090/bull/1820. URL https://www.ams.org/bull/2024-61-03/S0273-0979-2024-01820-5/.

Tomaso Poggio, Andrzej Banburski, and Qianli Liao. Theoretical issues in deep networks. *Proceedings of the National Academy of Sciences*, 117(48):30039–30045, December 2020. doi: 10.1073/pnas.1907369117. URL https://www.pnas.org/doi/10.1073/pnas.1907369117. Publisher: Proceedings of the National Academy of Sciences.

M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature neuroscience*, 2:1019–1025, 1999. tex.date-added: 2016-05-06 20:19:53 +0000 tex.date-modified: 2016-05-06 20:23:30 +0000.

Werner M. Seiler. *Involution: The formal theory of differential equations and its applications in computer algebra*, volume 24 of *Algorithms and computation in mathematics*. Springer Berlin Heidelberg, Berlin, 2010. ISBN 978-3-642-01286-0. doi: 10.1007/978-3-642-01287-7.

Herbert A. Simon. The Architecture of Complexity. In George J. Klir, editor, *Facets of Systems Science*, pages 457–476. Springer US, Boston, MA, 1991. ISBN 978-1-4899-0718-9. doi: 10.1007/978-1-4899-0718-9_31. URL https://doi.org/10.1007/978-1-4899-0718-9_31.

Elliott Sober. *Ockham's razors: a user's manual*. Cambridge University Press, Cambridge, 2015. ISBN 978-1-107-06849-0 978-1-107-69253-4.

Eduardo D. Sontag. Input to State Stability: Basic Concepts and Results. In Andrei A. Agrachev, A. Stephen Morse, Eduardo D. Sontag, Héctor J. Sussmann, Vadim I. Utkin, Paolo Nistri, and Gianna Stefani, editors, *Nonlinear and Optimal Control Theory: Lectures given at the C.I.M.E. Summer School held in Cetraro, Italy June 19–29, 2004*, pages 163–220. Springer, Berlin, Heidelberg, 2008. ISBN 978-3-540-77653-6. doi: 10.1007/978-3-540-77653-6_3. URL https://doi.org/10.1007/978-3-540-77653-6_3.

Appendix A. Hiearchy

In our setting, *hierarchy* is the structured way to decompose a (local) symmetry action and its induced state geometry.

Algebraic factorization (compositional generators). Locally (near the identity) we write each world transformation as an ordered product

$$\gamma \;=\; \gamma^{(L)}\gamma^{(L-1)}\cdots\gamma^{(1)}, \qquad \gamma^{(k)} \in G_k, \tag{10}$$

where each G_k is a (finite-parameter) Lie sub-pseudogroup generated by a small set of "level-k" infinitesimal symmetries. This is the formal expression of compositionality: complex deformations are built by nesting a few primitive moves close to the identity. A concrete instance is the product-of-exponentials model in robot kinematics, where each joint contributes a one-parameter subgroup and the chain is an iterated semidirect product (Lynch and Park, 2017).

Flag of sub-pseudogroups (omitting generators). Choosing the order in (10) induces a flag

$$G \;=\; H_0 \;\supset\; H_1 \;\supset\; \cdots \;\supset\; H_L, \qquad H_k \;:=\; G_L G_{L-1}\cdots G_{k+1}, \tag{11}$$

obtained by successively *omitting* generators. Each step $H_{k-1} \to H_k$ declares a new set of symmetry directions "already explained" at coarser scale. The quotients H_{k-1}/H_k collect the *residual* directions that remain to be explained at level k.

Geometric picture: orbit leaves and nested quotients. The action of H_k on the agent's state space defines a foliation by H_k-orbits. Passing to the orbit space produces a nested sequence of *reduced manifolds*

$$\mathcal{M}_0 \;\supset\; \mathcal{M}_1 := \mathcal{M}_0/H_1 \;\supset\; \cdots \;\supset\; \mathcal{M}_L := \mathcal{M}_{L-1}/H_L, \tag{12}$$

which we interpret as the agent's *hierarchical state representation*. Under static inputs, world-tracking fixes the Noether-style labels associated with the omitted generators, so trajectories remain on the corresponding orbit leaf; Eq. (12) formalizes our "nested invariant manifolds." The Spencer complex provides the compatibility conditions ensuring that these layered constraints are integrable (no hidden obstructions when one moves across levels) (Goldschmidt, 1967; Seiler, 2010).

Appearance/semantic parameters are part of the symmetry. In our generative view on "cat", *all* controllable attributes—pose and articulation, shape, illumination, texture, reflectance, eye/fur color, etc.—live in the configuration space $\mathcal{C}$ and are acted upon by the same Lie pseudogroup G. Equivalently, we may (locally) decompose

$$\mathcal{C} \;\simeq\; \mathcal{C}_{\text{pose/artic}} \times \mathcal{C}_{\text{shape}} \times \mathcal{C}_{\text{appearance}}, \qquad G \;\simeq\; G_{\text{pose}} \ltimes G_{\text{shape}} \ltimes G_{\text{appearance}},$$

and require the generative map to be equivariant with respect to the full action,

$$f(\gamma \cdot c) \;=\; \rho(\gamma) \cdot f(c), \qquad \gamma \in G. \tag{13}$$

Here ρ is the induced (generally nonlinear) action on image space, which covers not only geometric moves but also appearance changes (color/texture fields, reflectance, etc.). Thus,

the same hierarchical flag (11) may interleave pose-, shape-, and appearance-level generators; when a subgroup in the flag is "frozen," its labels are fixed and the system descends to the corresponding orbit quotient. For purely discriminative tasks (e.g., "cat/not-cat"), one may *choose* to quotient out some appearance factors at the top (yielding invariance). For world-tracking, the agent typically remains *equivariant* to all of them, so that $p(x) \approx I_{\theta(t)}$ follows those changes.

When inputs vary slowly or the internal hypothesis updates, level k produces a small *residual transformation* in the quotient, $\varepsilon_k \in H_{k-1}/H_k$, with $\varepsilon_k = \exp(\eta_k^a S_k^a)$, where $\{S_k^a\}$ span the Lie directions present in H_{k-1} but absent from H_k, and η_k^a are the *error coordinates*. These residuals induce drift/updates *along $\mathcal{M}_k$*.

The hierarchical construction above does not require coarse-graining. In practice, however, after canonicalizing the incoming signal by the current prediction, it is often useful to apply a task-dependent coarse-graining operator before passing the residual upward, to improve noise robustness and ensure that each level sees the information relevant at its scale. This "irons out" small mismatches without altering the orbit-quotient logic.

Appendix B. Generative vs. Predictive Hierarchies

In the Lie–pseudogroup formulation, hierarchy has two complementary directions: a *generative* (synthesis) order that goes from coarse to fine, and a *predictive* (inference) loop in which predictions flow top–down while errors flow bottom–up.

Near the identity, we factor world transformations as

$$\gamma = \gamma^{(L)}\gamma^{(L-1)}\cdots\gamma^{(1)}, \qquad \gamma^{(k)} \in A_k, \tag{14}$$

with a decreasing flag of cumulative remainders

$$H_k := A_{k+1}\cdots A_L, \qquad H_0 = G, \quad H_L = \{e\}, \tag{15}$$

so the new directions at level k are (locally) $A_k \simeq H_{k-1}/H_k$. Quotienting by H_k yields the nested reduced manifolds

$$\mathcal{M}_0 \supset \mathcal{M}_1 := \mathcal{M}_0/H_1 \supset \cdots \supset \mathcal{M}_L := \mathcal{M}_{L-1}/H_L. \tag{16}$$

Generative (synthesis) order: coarse → fine. To *generate* an instance from I_0, first fix the coarse orbit/quotient labels (which leaf of $\mathcal{M}_1, \mathcal{M}_2, \dots$), then compose the finer coset moves in order:

$$I = \gamma \cdot I_0 = \left(\gamma^{(L)} \cdots \gamma^{(1)}\right) \cdot I_0, \qquad \gamma^{(k)} \in A_k \simeq H_{k-1}/H_k. \tag{17}$$

Predictive (inference) loop: predictions down, errors up. At time t level k carries $\hat{\gamma}_k(t) \in H_k$ and issues a top–down prediction $\hat{I}_k = \hat{\gamma}_k \cdot I_0$. The mismatch with the datum $I_{\theta(t)}$ is explained (for small errors) by a bottom–up residual in the quotient,

$$\varepsilon_k \in H_{k-1}/H_k \simeq A_k, \qquad \varepsilon_k = \exp(\eta_k^a S_a^{(k)}), \tag{18}$$

estimated by linearising the action at $\hat{I}_k$. Level k updates with the part it can realise inside H_k and passes only the unresolved quotient directions upward (e.g. $\eta_k = P_k\eta_k + Q_k\eta_k$, send $Q_k\eta_k$).

Static vs. adiabatic. With static inputs, equivariance plus tracking fix the Noether-style labels of the omitted generators, so trajectories remain on a leaf $\mathcal{M}_k$. Under slow drift, these labels are adiabatic invariants and residuals induce controlled motion *within* $\mathcal{M}_k$.

Appendix C. From Lie Hierarchy to a Blender Rig (Flow-style Cat)

A production rig in the software BLENDER (Community, 2018) provides a concrete realisation of our Lie-pseudogroup hierarchy for a cat character (as in a scene of the Blender-generated movie *Flow*). Let $\mathcal{C}$ collect all controllable model parameters (pose, articulation, shape, groom, materials, lighting, camera), and let $f : \mathcal{C} \to \mathbb{R}^X$ render an image (Cycles/Eevee). We arrange the local action of G near the identity as a factorisation $\gamma = \gamma^{(L)} \cdots \gamma^{(1)}$ with $\gamma^{(k)} \in A_k$, and define the remainder flag $H_k := A_{k+1} \cdots A_L$ (§A). The table below maps levels to standard Blender constructs (see also Figure 3):

Level k	Generators A_k (local)	Blender construct (examples)
1	Camera $SE(3)$ and lens	`camera.matrix_world`, focal length, sensor
2	Global body/root $SE(3)$	armature root bone (`pose.bones["root"]`)
3	Torso/spine chain (PoE)	spine bones, FK/IK; constraints, drivers
4	Limbs/paws/tail (PoE)	limb bones; IK solvers; stretch-to constraints
5	Facial morphology	shape keys (blendshapes), jaw/ear bones
6	Fur/appearance	material node params (albedo/roughness); groom P.C.s
7	Illumination gauge	light transforms; spherical-harmonic coeffs
8	Environment/camera jitter	world rotation; rolling shutter, exposure

Compositional action (PoE). Rigid/articulated motion uses the product-of-exponentials (PoE) (Lynch and Park, 2017):

$$T(\theta) \;=\; \Big(\prod_{n\in\text{chain}} e^{[S_n]\theta_n} \Big) M, \tag{19}$$

with twists $[S_n] \in \mathfrak{se}(3)$ for bones and M the bone's home pose. This is exactly the ordered composition $\gamma^{(k)} \in A_k$ at the geometric levels (2–4). Appearance and lighting levels act by nonlinear but smooth pseudogroup transformations on shader/groom/light parameters; the induced action $\rho(\gamma)$ on image space remains local (§2).

Orbit quotients and nested manifolds. Freezing levels up to k (i.e., quotienting by H_k) collapses all configurations equivalent under those generators, yielding the reduced leaf $\mathcal{M}_k = \mathcal{M}_{k-1}/H_k$. For a static shot, world-tracking pins the Noether labels of the omitted generators, so the render/feature readouts stay constant on $\mathcal{M}_k$ (cf. Eq. (12)).

Predictive residuals in practice. At frame t level k predicts $\hat{I}_k = \hat{\gamma}_k \cdot I_0$ (rendered from $\hat{c}_k$). The error $\delta I_k = I_{\theta(t)} - \hat{I}_k$ is explained by a small residual $\varepsilon_k \in H_{k-1}/H_k \simeq A_k$: linearise the image change along the level-k generators,

$$\delta I_k \;\approx\; \sum_a \eta_k^a V_a^{(k)}(\hat{I}_k), \qquad \varepsilon_k \simeq \exp(\eta_k^a S_a^{(k)}), \tag{20}$$

where $V_a^{(k)}$ are image-space velocities induced by basis $S_a^{(k)} \in \mathrm{Lie}(A_k)$ (estimated by finite-difference renders). Update $\hat{\gamma}_k \leftarrow \hat{\gamma}_k \exp(P_k \eta_k)$ using the part $P_k \eta_k$ realisable within H_k, and pass the unresolved component $Q_k \eta_k$ upward (§3.2).

Blender blueprint (schematic). *Parameterisation.* A minimal state vector can include: camera $(6 + \mathrm{lens})$, root (6), $\sim$20–40 joint angles (PoE), 10–30 shape-key coeffs, 3–8 groom principal components, 9 SH lighting coeffs; the exact split defines the sets A_k. *Application.* Within bpy, each $\gamma^{(k)}$ is applied by writing pose bone transforms, shape key values, material node sockets, and light/world transforms, then rendering to a buffer:

$$\hat{I}_k = f\big(\gamma^{(k)} \cdot \hat{c}_{k-1}\big).$$

Spencer/compatibility as rig constraints. Blender's dependency graph (constraints, drivers, IK) enforces integrability of the layered actions—precisely the "no hidden obstruction" condition captured abstractly by the Spencer complex (Seiler, 2010). In practice: no cycles in drivers; consistent bone/rest matrices; shader/groom parameters driven only by lower levels or constants.

Why this matches the movie-making intuition. The director's controls are inherently hierarchical: camera and blocking first, then body motion, then paws/tail, then facial nuance, then appearance/lighting. Our $G = H_0 \supset \cdots \supset H_L$ flag formalises that workflow: each stage *defines* a quotient leaf $\mathcal{M}_k$ (solid arrows in Fig. 2), while residuals H_{k-1}/H_k *move* the solution along that leaf (dashed arrows), until the frame's constraints are met (static case) or track a slowly varying target (adiabatic case). The PoE levels (2–4) give exact Lie-group composition; appearance/lighting act as smooth pseudogroup charts.
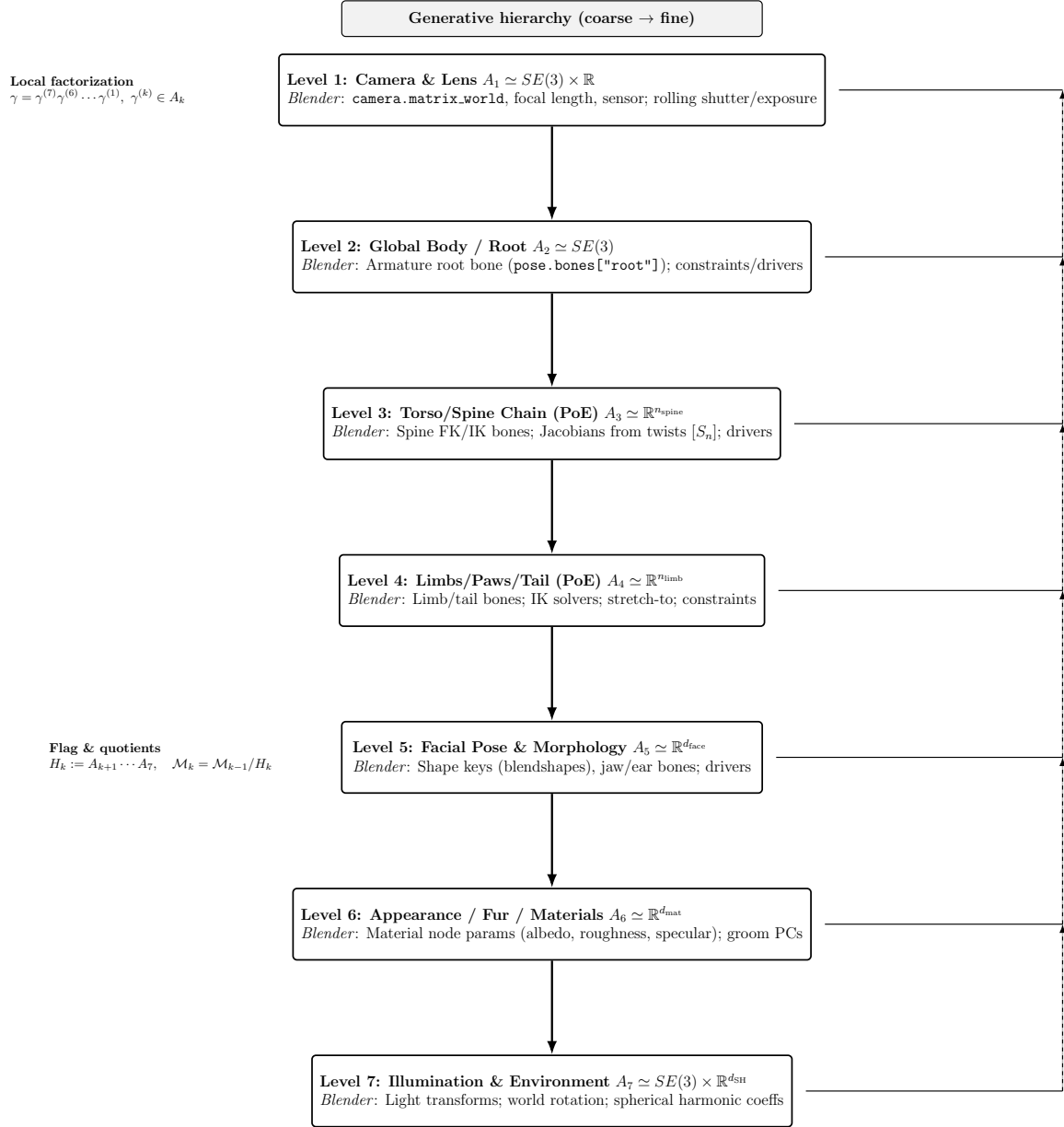
Figure 3: Blender cat example hierarchy as a Lie-pseudogroup ladder. Solid arrows (left) show the generative order (coarse → fine). A right-hand fine → coarse *error rail* aggregates bottom-up prediction residuals (dashed arrows) with short connectors from each level. Each level A_k is a local group/pseudogroup factor; the flag $H_k := A_{k+1} \cdots A_7$ induces nested quotients $\mathcal{M}_k$.