Representing Part-Whole Hierarchy with Nested Neuronal Coherence

Hao Zheng[†] **Hui Lin** **Rong Zhao**[*]
Center for Brain-Inspired Computing Research,
Optical Memory National Engineering Research Center,
Tsinghua University - China Electronics Technology HIK Group
Co. Joint Research Center for Brain-Inspired Computing,
IDG / McGovern Institute for Brain Research at Tsinghua University,
Department of Precision Instrument,
Tsinghua University, Beijing 100084, China.

Abstract

Human vision flexibly extracts part-whole hierarchies from visual scenes, but representing such structures remains a key challenge for neural networks. Inspired by the neural syntax hypothesis in neuroscience, we propose a framework for representing hierarchical part-whole relationships through nested neuronal coherence, characterized by its continuous and distributed nature. Nested neuronal coherence refers to dynamical states where distributed activations are temporally correlated to form cell assembly sequences across timescales, argued to represent neural syntax. Building on this framework, we develop a cortex-inspired hybrid model, Composer, which dynamically achieves emergent nestedness when given images. To evaluate the emergent hierarchy, we create four synthetic datasets and three quantitative metrics, demonstrating the model's ability to parse scenes of varying complexity. Overall, our work advances a systematic approach spanning representation, implementation, and evaluation toward reconciling the discrete, compositional structure of part-whole hierarchies with the continuous, distributed representations of neural networks.

1 Introduction

Psychological evidence suggests that humans impose part-whole hierarchies to manage complexity, organizing objects at one level as compositions of objects at the next [1–3]. Such compositional structure is also essential for interpreting neural representations. However, representing hierarchical structure in artificial neural networks remains a profound challenge [4], as illustrated below.

A representational scheme for handling part-whole relationships in connectionist models must, at a minimum, specify (1) how distributed activations are grouped into individual entities at each level (wholes and parts), (2) how the meanings of wholes relate to those of their constituents, and (3) how the entire structure (parts, wholes, and their relationships) can be simultaneously represented as neural activations [5]. While recent advances in object-centric representations offer insights into single-level grouping [6–14], little progress has been made in specifying hierarchical relationships or how wholes, parts, and their relations are jointly encoded as neural activations, leaving the broader challenge largely unresolved.

At the computational level, this challenge is illustrated by an example: given a visual scene (Fig. 1a), the mental interpretation of both the compositional structure and the meanings of constituents can

[*]Corresponding author: r_zhao@mail.tsinghua.edu.cn.
[†]Now at Massachusetts Institute of Technology, Cambridge, MA, USA. Email: hao_z@mit.edu.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

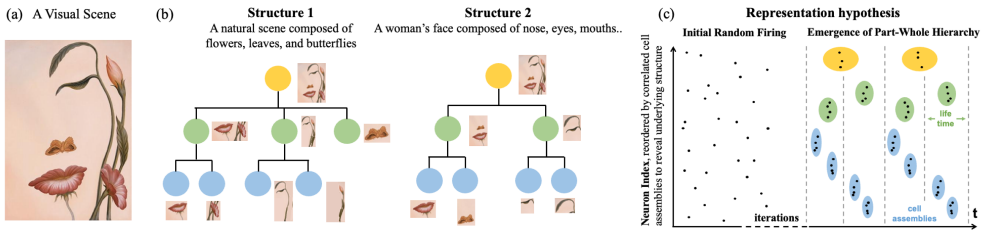

Figure 1: (a) A visual scene. Adapted from *Mouth of Flowers* by Octavio Ocampo. (b) Different plausible mental parsing structures of the same scene. (c) Schematic spike raster plot (y-axis: reordered neuron index; x-axis: time; dashed x-axis denotes skipped steps): representing structure 2 in (b) as nested neuronal coherence (right), emerging from initial random firing (left). Cell assemblies (right) are highlighted by colored shadows. Color indicates levels in all cases.

be uncertain—such as the number, grouping, and relations of objects. Thus, the structure must be dynamically inferred from the "content" and continuously constrained by the self-consistency of multiple inconclusive cues [15]. This uncertain, content-sensitive nature of part-whole hierarchies challenges solutions directly inspired by symbolic processing, such as slot-based models [6, 11, 16], which are discrete and require predefining structure independently of content.

A third challenge at the evaluation level is more technical but equally critical: there is a lack of datasets explicitly annotating part-whole hierarchies and no precise metric for assessing hierarchical structure in neural network latent spaces. As a result, most neural network models are demonstrated on datasets without clear hierarchies [16, 17] and validated by less relevant metrics such as classification accuracy [17, 18] or single-level segmentation scores [16]. Conclusions about hierarchical structure are often based on limited visualizations, leaving open whether the models truly fulfill their claims.

In this paper, inspired by the "Neural Syntax Hypothesis" in neuroscience [19], we formulate discrete part-whole hierarchies as continuous correlation structures among contents (nested neuronal coherence), with the inference of constituents and structure realized as a unified dynamical process converging to a final state based on the self-consistency between content and structure from initial uncertainty. For evaluation, we develop synthetic datasets with explicit part-whole structures and propose metrics that comprehensively assess hierarchical structure quality, covering both constituents and their relations. Finally, we empirically validate our framework by showing that our model outperforms the state of the art on this problem. Our contributions are summarized as follows:

(1) We propose a brain-inspired framework to address the part-whole hierarchy in artificial neural networks through continuous neuronal coherence instead of discrete slots.

(2) We introduce **Composer** (short for **CO**rtical-inspired e**M**ergence of **P**art-wh**O**le relation**S**hip through n**E**uronal cohe**R**ence), a brain-inspired model combining biological constraints and machine learning, to demonstrate the inference of the part-whole hierarchy as an emergent phenomenon.

(3) We develop datasets and metrics to explicitly reveal hierarchical structure from uncertain, distributed neural representations. Taken together, we pioneer a systematic framework for part-whole hierarchies in neural networks—spanning representation, implementation, and evaluation—to inspire advances toward bridging human and machine vision.

2 Neuroscience-Motivated Representational Hypothesis

A widely discussed hypothesis in neuroscience proposes that transiently active ensembles of neurons, known as "cell assemblies," represent distinct cognitive entities [20]. These assemblies are thought to be dynamically correlated and organized into sequences [20] (Fig. 1c), accompanied by gamma-band rhythms [21]. Such oscillatory firing patterns are posited to convey two types of messages: (1) discharge frequency encodes feature presence, and (2) spike timing encodes feature binding [22, 23]. Although this view remains debated [24] and alternative binding mechanisms have been proposed [25, 26], it offers a natural account of dynamic grouping: temporally correlated spikes "bind" distributed features (contents) into entities, or neural words (structural units). Furthermore, the hierarchical organization of assembly sequences may be regarded as a neural syntax, with nested part-whole relationships represented as temporally nested structures of assembly trajectories, accompanied by nested rhythms [19] (Fig. 1c). These hypotheses motivate a coherence-based representation paradigm for part-whole hierarchies, introduced as follows.

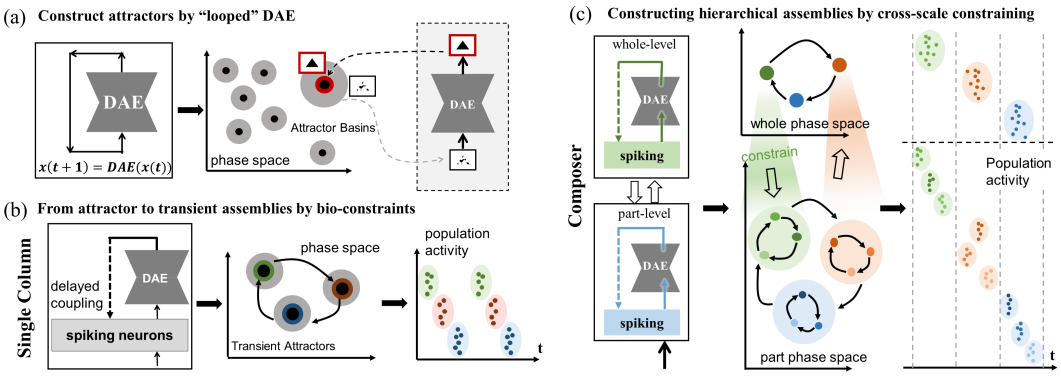

Figure 2: How nested neuronal coherence emerges: an intuitive explanation. (a) A denoising auto-encoder (DAE, right) forms attractors (middle) by connecting its input and output layers (left). (b) Single-level model: Refractoriness and coupling delays destabilize stable attractors into transient ones (middle), supporting rhythmic coherence (right). (c) Multi-level Composer: Hierarchically nested sequences (middle), underpinning nested neuronal coherence (right), emerge via cross-level constraints between columns operating at different timescales (left and middle).

Discrete, spatially aligned levels. Given a visual scene (Fig. 1a), the part-whole hierarchy is represented across distinct *discrete* levels, each supported by a layer of spiking neurons. All levels share the same topographic spatial layout as the input image [4]. We fix a small number of levels as a simplification, since a limited set of representational levels is argued to suffice for human visual understanding [4]; the design extends naturally to more levels (Appendix A.8.6).

Concurrent representation and grouping via multi-scale correlations. Unlike feature hierarchies in CNNs, where successive layers extract increasingly abstract features, every level concurrently represents the features of the entire scene, and levels differ only in how these features are grouped [4]. Feature neurons at different levels form correlation patterns across distinct spatial-temporal scales, organizing synchronized cell assemblies of varying sizes that group features into parts or wholes (Fig. 1c, right, cell assemblies).

Assembly sequence hierarchy as part-whole hierarchy. Whole-level assemblies persist longer and temporally embed sequences of part-level assemblies, dynamically forming nested structures that mirror part-whole hierarchies as neural activations (Fig. 1c, right).

3 Intuition of the Mechanism

How can a hierarchically structured assembly sequence emerge from randomness, with parts temporally nested within wholes (Fig. 1c)? This challenge involves two subproblems: (1) how assembly sequences at each level form neural words (parts/wholes), and (2) how assemblies across levels nest to create neural syntax (hierarchical relationships).

To generate assembly sequences, we combine positive and negative feedback: a cell assembly is transiently activated by positive feedback and quickly suppressed by negative feedback, enabling sequential emergence. The recurrent DAE provides positive feedback that stabilizes firing patterns, while biological constraints — spiking, neural adaptation, and transmission delays — drive switching between assemblies.

From DAE to attractor networks. Positive feedback arises naturally from the connection between denoising auto-encoders (DAE, [27]) and attractor networks (Fig.2a). Training a DAE on a large dataset and connecting its input and output layers yields a dynamical system $x_{t+1} = DAE(x_t)$, where each data sample becomes an attractor under "bottom-up/top-down" dynamics (Fig.2a, middle). Denoising acts as positive feedback, stabilizing associative memories parameterized within the DAE [28].

From attractors to rhythmic assembly sequences. Negative feedback is introduced through neuronal adaptation, a ubiquitous biological constraint. As shown in Fig.2b (left), a layer of spiking neurons with refractory periods connects the DAE output and input (plus external input). Refractoriness destabilizes attractors, making them transient. Coupling delays further introduce switching

windows between alternative assemblies (Fig.2b, middle). Together, these mechanisms transform stable attractors into cyclic sequences, underpinning rhythmic coherence states (see [14] for theoretical proof).

From assembly sequence to assembly hierarchy. To induce hierarchy, we leverage integration time windows. Buzsaki [19] proposed that nested time windows serve as the "syntax" of cell assemblies, providing a bias toward hierarchical composition in the brain, aligning with cortical gradients of timescales [29]. In Fig.2c (left), multiple DAE-based modules ("columns") are stacked. Higher-level columns integrate over longer windows, coordinating slower switching dynamics, while lower levels integrate faster. Activated long-lived assemblies at higher levels gate lower-level dynamics, while integrated lower-level activities drive higher-level states (Fig.2c, middle). These constraints promote the emergence of nested neuronal coherence. In the following section, we detail the full Composer architecture.

4 Model

In this work, we focus on a two-level hierarchy (whole and part) and restrict analysis to the pixel level, enabling a tractable investigation of neuronal coherence. This framework can be extended to more levels in future work (see A.1). Overall, Composer consists of two columns interconnected by top-down gating and bottom-up integration (Fig.2c). Each column includes a pixel-level spiking layer, referred to as the spike coding space (SCS, Fig.3c), and a denoising autoencoder (DAE). "Pixel-level" indicates that the SCS at both levels matches the image dimensions (d). Within each column, the SCS and DAE are delay-coupled: delayed feedback from the DAE gates the firing rates of spiking neurons (consistent with dendritic computations in cortical pyramidal cells [30]). Conversely, spikes in the SCS are integrated over a short time window prior to entering the DAE, effectively serving as a coincidence detector [31]. In what follows, we first introduce the column architecture and then explain cross-level connections.

4.1 Part-level column

Spiking neurons in SCS_1 of the part column receive inputs from three sources (Fig.3b): the external input image $x \in \{0,1\}^d$, inner-column feedback $\gamma_1(t) \in \mathbb{R}^d$ and inter-column feedback $\Gamma(t) \in \mathbb{R}^d$ at each time step $t \in [0, T]$ (where d is the number of image pixels and neurons in SCS_1):

$$\rho_1(t) = x \cdot \gamma_1(t) \cdot \Gamma(t) \tag{1}$$

Here, '$\cdot$' denotes pixel-wise product and firing rate $\rho_1(t)$ determines the firing activity $s_1(t) \in \{0,1\}^d$:

$$P(s_1(t) = 1) = \rho_1(t) \cdot g_1(t - \hat{t}), \quad t \in [0, T]. \tag{2}$$

Here, $P()$ denotes the probability and $g_1(t-\hat{t})$ represents the relative refractory function of part-level neurons (Fig.3e), where $\hat{t}$ indicates the timing of the last spike firing event (for each neuron). After firing, the neuron enters an absolute refractory period of duration τ_{r1}, followed by a relative refractory period lasting $\tau_{\delta 1} - \tau_{r1}$, during which its firing probability is suppressed by a factor $g < 1$ (Fig.3e). The inner-column feedback $\gamma_1(t)$ in Eq.1 is the denoised output of DAE_1, delayed by τ_d:

$$\gamma_1(t) = \mathrm{DAE}_1([I_1 * s_1](t - \tau_d)). \tag{3}$$

where $*$ is the convolution operator and I_1 is the integrative function for $s_1(t)$ with a time window of τ_1 (Fig.3f): $[I_1 * s_1](t) = \sum_{\tau=0}^{\tau_1}(I_1(\tau) \cdot s_1(t-\tau))$. To clarify, the spiking activity $s_1(t)$ in the SCS is integrated over a short time window τ_1 before being fed into DAE_1, and the feedback from DAE_1 to SCS_1 is delayed by τ_d.

4.2 Whole-level column

As the top level in the two-level Composer, the whole-level column does not receive top-down modulation from higher levels. The firing rate $\rho_2(t)$ of spiking neurons in SCS_2 is determined by the product of a driving term and a modulation term:

$$\rho_2(t) = (\lambda \cdot x + \overline{\lambda} \cdot D(t)) \cdot \gamma_2(t) \tag{4}$$

where $\lambda < 1$ is the factor of partial influence from skip connection (Fig.3c) and $\overline{\lambda} = 1 - \lambda$. $D(t)$ is the integrated spikes from the part-level SCS_1, serving as the main driving input for the whole-level column. ρ_2 similarly determines the spike firing in SCS_2 by:

$$P(s_2(t) = 1) = \rho_2(t) \cdot g_2(t - \hat{t}) \quad (5)$$

Besides, γ_2 in Eq.4 is the delayed feedback from DAE_2 (similar to Eq.3):

$$\gamma_2(t) = \mathrm{DAE}_2([I_2 * s_2](t - \tau_d)) \quad (6)$$

4.3 Linking the levels

Up to now, we have introduced operations within each column of the Composer except for two variables: $\Gamma(t)$ in Eq.1 and $D(t)$ in Eq.4, which describe interactions between levels:

$$\begin{aligned} \Gamma(t) &= [I_\Gamma * s_2](t - \tau_{d'}), \\ D(t) &= [I_D * s_1](t). \end{aligned} \quad (7)$$

where I_Γ, I_D is the corresponding integration function. $\tau_{d'}$ is the cross-level feedback delay, which is set to be $\tau_{d'} = \tau_d$ for simplicity.

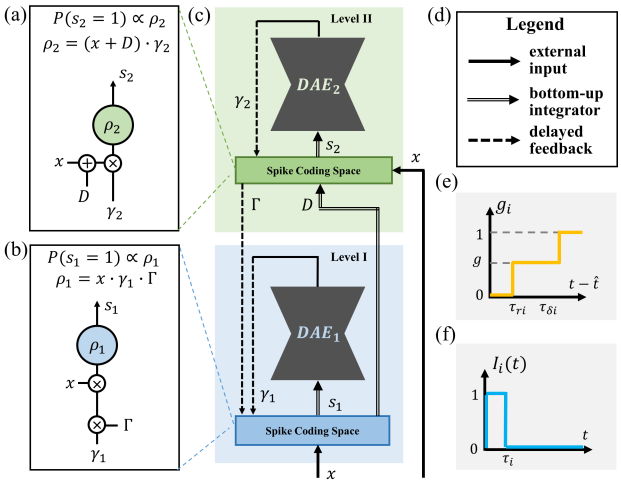

Figure 3: (a)(b) Pyramidal neuron models in the spike coding space of part level and whole level. $\otimes$ denotes multiplication and $\oplus$ indicates addition at the dendrites. (c) Detailed information flow. x is the external input. Note that delayed coupling exists both within each column and between different columns. Levels are color-coded. (d) Legend for (c). (e) Relative refractory function g_i. (f) Integration function $I_i(t)$ with timescale τ_i.

Each integration function I_i has its corresponding time window parameter τ_i ($i \in 1, 2, \Gamma, D$, Fig.3f), and each refractory function g_i is characterized by an absolute refractory period τ_{ri} within the total refractory period $\tau_{\delta i}$ ($i \in 1, 2$, Fig.3e). The general relationship among these parameters is: $\tau_1 < \tau_2 < \tau_D \sim \tau_\Gamma \ll \tau_d$ and $\tau_{\delta 1} \sim \tau_{\delta 2} \ll \tau_d$.

As suggested by neuroscience, hierarchical organization of integration time windows and delay coupling are key mechanisms for forming cell assemblies [19]. These time-scale parameters are treated as hyperparameters; for details and ablation results, see A.4.2. The model's performance is generally robust to variations within a reasonable range.

DAE_1 and DAE_2 are pre-trained on part- and whole-object datasets using a BCE loss, functioning as pre-configured wiring in cortical columns [32]. The DAEs thus carry level-specific semantic priors, determining what kind of content each level tends to represent (e.g., the appearance of parts versus wholes). They do not, however, specify the structure of a given scene, since no loss compares the representation with a target structure. Instead, the grouping of pixels, the number of assemblies, and their nesting across levels emerge dynamically through content-sensitive inference in the Composer. See A.5 for training details.

5 Evaluation

Quantitative evaluation of hierarchical structure is crucial for model validation but remains scarce in related works [16–18]. Two main challenges arise: (1) real-world images often lack explicit parsing or annotation, and (2) distributed hierarchical structures are difficult to measure. These challenges motivate the development of synthetic datasets and metrics for evaluating Composer.

5.1 Dataset

We created four synthetic part-whole datasets of varying complexities (Fig. 4), each containing 60,000 samples. Each image comprises multiple whole objects, each composed of clearly defined parts.

Ts (Fig. 4a) features three letters $\top$ and three reversed letters $\perp$, each assembled from a horizontal and vertical bar, resembling stimuli used in perceptual neuroscience tasks [33].

SHOPs (Fig. 4b) includes **S**hoes (ii), **H**ouses (i), and **OP**eras (iii), where each whole is assembled from elementary rectangles and triangles, with overlapping parts left unassigned due to belonging ambiguity.

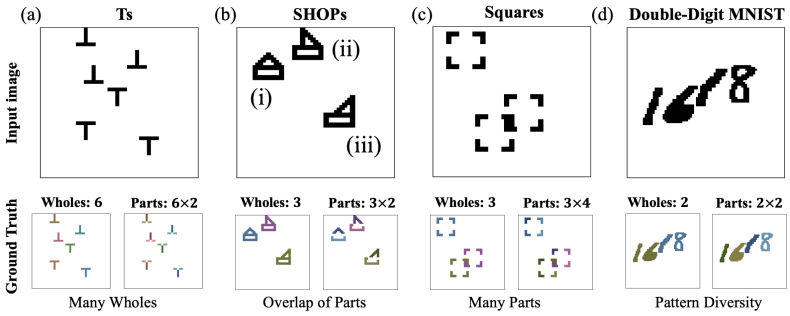

Figure 4: Dataset samples: (a) Ts; (b) SHOPs; (c) Squares; (d) Double-Digit MNIST. Top: input; Bottom: ground truth for Wholes (left) and Parts (right), with the counts of wholes and parts indicated above and the complexity for each dataset shown below.

Squares (Fig. 4c) contains three squares, each defined by four corners, with higher part complexity than Ts; similar patterns are used in Gestalt studies of illusory contours [34].

Double-Digit MNIST (Fig. 4d) features two randomly placed double-digit objects, each formed by two MNIST digits, resulting in greater complexity and diversity.

5.2 Scores

How can we evaluate the nestedness shown in Fig. 1c (right)? Our approach builds on the idea that correlated cell assemblies can be conceptualized as clusters of spike trains. Thus, coherence measures for clustering, such as the Silhouette Score [35], are natural candidates.

We introduce the **Part Score** and **Whole Score** to quantify neuronal coherence within each level:

$$\text{Part Score} = \text{Silhouette}(s_1, s_1, \text{GT}_1, \text{D}_{\text{vp}}) \tag{8}$$

$$\text{Whole Score} = \text{Silhouette}(s_2, s_2, \text{GT}_2, \text{D}_{\text{vp}}) \tag{9}$$

where $s_1, s_2 \in \{0,1\}^{d \times \tau_l}$ represent spike trains of d neurons over τ_l time steps at two levels (indicated by subscripts 1 and 2), and GT_1 and GT_2 are ground-truth labels for parts and wholes, and D_{vp} is the Victor-Purpura metric [36] used to measure spike train dissimilarity. Since spiking neurons in the SCS have a one-to-one correspondence with image pixels (Fig. 5a), ground-truth labels can be directly applied to spike trains. These labels are used solely for evaluation, not for training. Lastly, the Silhouette function $\text{Silhouette}(a, b, L, D)$ measures the inner-cluster coherence between two sets of samples, a and b (typically $a = b$), based on cluster assignments L (assigning each sample a label $l \in L$) and a distance metric D (measuring pairwise distances). Higher values reflect tighter clustering, implying stronger synchrony among spikes encoding the same part or whole.

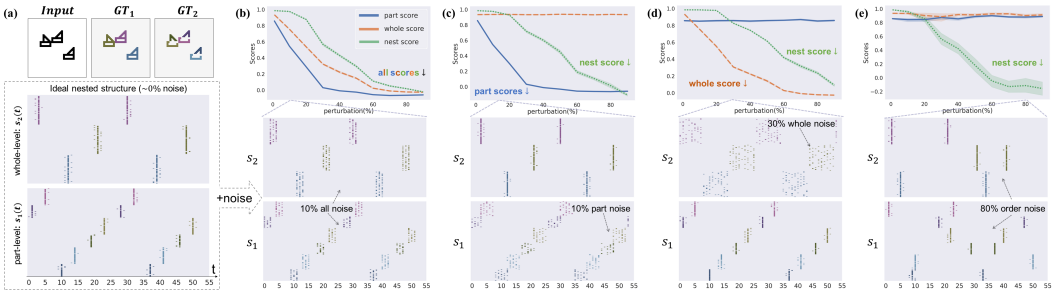

Figure 5: **Validation of Scores.** (a) Top: hypothetical input; bottom: ideal two-level nested spike pattern constructed from the input, ground truth, and the representational hypothesis. (b–d) Adding uncorrelated spike timing noise at both levels (b), part level (c), or whole level (d) progressively degrades the corresponding scores. (e) Group-wise correlated noise (identical within group, varied across groups) selectively disrupts the part–whole alignment while preserving intra-group structure, and results in degradation of the Nest Score only, validating its sensitivity to hierarchical structure. (b–e) Top: score values vs. noise level; bottom: example of a perturbed spike pattern at a selected noise level (details in Fig. 13–16). Each raster row is a neuron arranged topographically by the input image; colors indicate the corresponding object (cf. GT), and vertical alignment reflects synchrony.

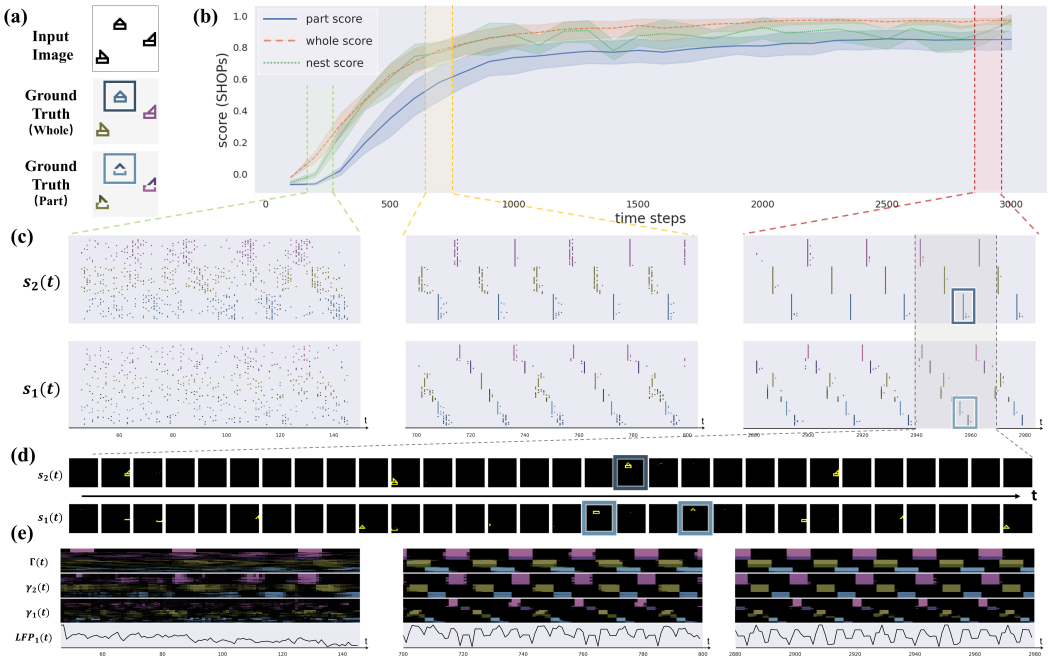

Figure 6: Emergence of the part-whole hierarchy with nested neuronal coherence. Exemplified by one SHOPs sample. (a) Input image and ground truth; (b) Evolution of the Scores. (c) The spike raster plot of three selected phases in (b): phase I (initial, green box), phase II (middle, yellow box), phase III (final, red box). $s_2(t), s_1(t)$ stand for spiking representations in SCSs of whole/part levels. (d) zoomed in spiking pattern during the period marked by black box in (c), to visualize what each cell assembly represents (e.g. blue boxes in (c),(d) and (a)). (e) Evolution of the top-down attention maps— $\Gamma_1(t), \gamma_2(t), \gamma_1(t)$, and the local field potential (LFP_1) during the three phases in (b).

To evaluate cross-level nestedness, we define a **Nest Score**, which generalizes Silhouette to measure coherence between s_1 and s_2 under the whole-level label GT_2 :

$$\text{Nest Score} \propto \text{Silhouette}(s_1, s_2, \text{GT}_2, \text{D}_{\text{vp}}) \tag{10}$$

This reflects the proximity between part-level and whole-level spike trains, capturing the nested structure (see A.3.4 for details).

We validate the scores' behavior by systematically adding noise to spike timing in a manually crafted ideal nested structure (Fig. 5a), generating a series of perturbed structures with varying noise levels. Perturbing only parts decreases the Part Score and Nest Score while leaving the Whole Score stable (Fig.5c); perturbing wholes affects the Whole and Nest Scores but not the Part Score (Fig.5d). Shuffling the relative timing between parts and wholes selectively reduces the Nest Score, with Part and Whole Scores stable (Fig.5e).

Collectively, the three scores comprehensively characterize nested neuronal coherence, with 1 indicating perfect coherence, 0 randomness, and -1 incoherence.

6 Experiments

6.1 Qualitative results and visualization

Emergence of the "Neural Syntax" in SCS. Fig.6 illustrates a simulation on a random sample from the SHOPs dataset. The convergence of the Part, Whole, and Nest Scores (Fig.6b) indicates that Composer progressively establishes neuronal coherence, representing parts and wholes as synchronized cell assemblies. In the final phase III (Fig.6c, right), a periodic emergence of three two-level binary trees is observed, corresponding to the three SHOPs objects (Fig.6a). One such tree is highlighted with deep blue (whole object) and light blue (part objects) boxes. Spikes are sorted and colored by ground truth to aid visualization. Notably, the structured representation (Fig.6c, right) continuously emerges from initial uncertainty (Fig.6c, left) through self-consistency among contents. Additional visualizations are provided in A.8.4.

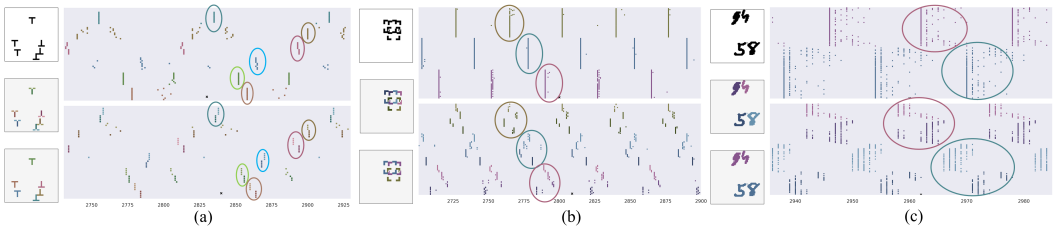

Figure 7: Visualization of the emergent part-whole hierarchy on other datasets: (a) Ts (b) Squares (c) Double-Digit MNIST. (In each sub-figure) Left: input image, part-level ground truth and whole-level ground truth. Right: spike raster plot in final phase III, top for whole level and bottom for part level. Cell assemblies are circled for clarification. See Fig.22∼Fig.29 for complete visualizations.

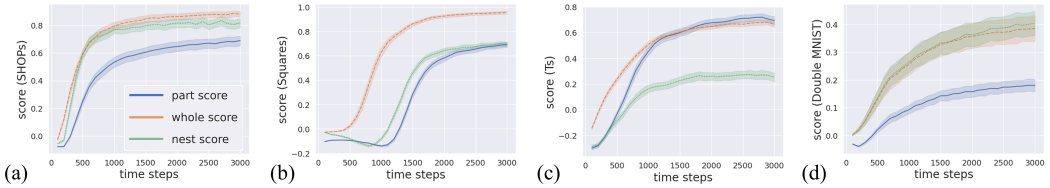

Figure 8: Convergence of Scores. (a) SHOPs (b) Squares (c) Ts (d) Double-Digit MNIST.

Emergence of the Hierarchical Attention Sequence. Fig.6e shows that starting from random activity (left), the cross-level feedback Γ and intra-level feedbacks γ_1, γ_2 progressively converge to a sequential trajectory. The close similarity between the assembly sequence (Fig.6c) and the attention sequence (Fig.6e) suggests a strong link between predictive top-down feedback and oscillatory assemblies, consistent with neuroscience findings [37]. Together, the DAE's top-down attention and the SCS's bottom-up spike integration facilitate the concurrent emergence of the attention sequence, neuronal coherence, and rhythmic activity (denoted as LFP_1, Fig.6e).

Visualization Results on Other Datasets. Fig. 7 shows the emergent coherence structures across different datasets. In the Ts dataset, six binary trees emerge periodically; in the Squares dataset, three quadtrees emerge; and in the Double-Digit MNIST, two binary trees emerge. Overall, Composer effectively represents the part-whole hierarchies of scenes with varying complexity.

6.2 Quantitative analysis

Adapting to diverse scene structures. To test whether the emergent structure adapts to the scene, we vary the number of objects in SHOPs scenes from 2 to 5. The same Composer, with identical DAEs and parameters, is applied to all scenes (Table 1). Without retraining, the number of assemblies and their nesting change with the scene, and scores degrade only gradually (Nest Score from 0.906 to 0.602), with the largest drop at five objects, where heavy overlap on the fixed canvas makes the scene itself ambiguous. Thus, while the levels are fixed by design, the number, membership, and nesting of structural units are inferred from the input rather than pre-assigned as in slot-based models, illustrating the flexibility gained by representing structure continuously. See A.8.2 for more details.

Table 1: Scores with varying number of SHOPs objects (mean over 1000 samples, 5 seeds; all std < 0.01).

#obj	2	3	4	5
Part	.678	.672	.517	.358
Whole	.914	.867	.775	.636
Nest	.906	.824	.763	.602

Score Convergence. Fig. 8 shows score convergence across iterations using 100 random samples per dataset. Scores converge consistently with negligible error bars across all datasets, though the convergence dynamics vary: In Squares (Fig. 8b), whole objects group faster than parts, mirroring human visual processing [34]; in Ts (Fig. 8c), the large number of objects places higher demands on cross-level coordination, resulting in a lagging Nest Score; in Double-Digit MNIST (Fig. 8d), distinguishing part-level MNIST digits is more challenging due to greater diversity. See A.8.1 for further details.

Benchmarking. We compare the Composer with a state-of-the-art model, the Agglomerator [17], which leverages neuronal coherence (*islands of identical vectors*) to group neuronal representations into a hierarchical structure. Using 1000 random samples and 5 random seeds, we evaluate both

models across the four datasets. We extend the coherence-based metric to assess the Agglomerator. As shown in Fig. 9, the Composer consistently outperforms the Agglomerator on all datasets. Notably, the Agglomerator fails to generate object-centric representations at each level, raising concerns about the reliability of its parsing capabilities. See Section A.7 for details and further explanation.

Ablation Study 1: DAE vs Coherence. Since denoised feedback from the DAE is critical in the Composer, we investigate the relationship between denoising efficacy and coherence scores. We pre-trained 100 DAE pairs on the SHOPs dataset—each consisting of a part-level and a whole-level DAE—using identical architectures but varying learning rates to produce a range of denoising losses. These pre-trained DAEs were integrated into the Composer for parsing tasks. Fig. 10a shows a strong correlation between lower DAE denoising loss and higher coherence scores across all metrics, suggesting that the effectiveness of denoising feedback is directly linked to the quality of nested coherence.

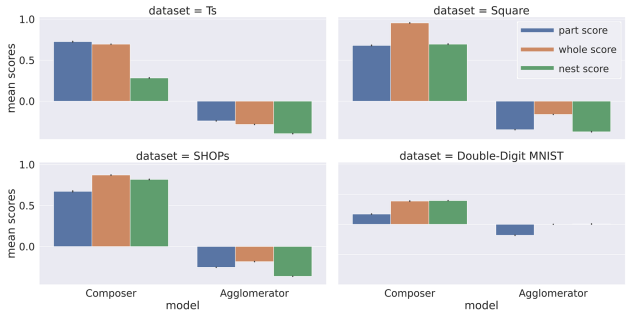

Figure 9: Benchmarking results on four datasets

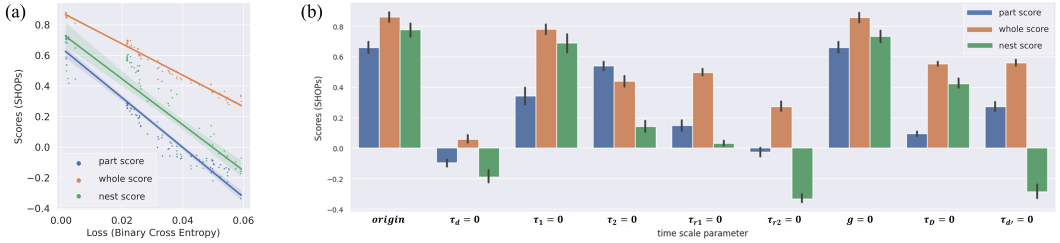

Figure 10: Ablation study on SHOPs. (a) Loss vs Scores (b) Ablation of time scale parameters. See Fig.17, Fig.20 for more results.

Ablation Study 2: Timescale Parameters. A key component of the Composer is a set of bio-inspired constraints, implemented via time-scale parameters. We assess their contribution by ablating each parameter individually (setting it to zero) on the SHOPs dataset (Fig. 10b). For example, $g = 0$ eliminates the relative refractory period, and $\tau_d = 0$ removes the inner-column feedback delay. Compared to the original model, all ablated variants show reduced scores. Notably, removing the inner-column feedback delay (τ_d), the whole-level refractory period (τ_{r2}), or the cross-level feedback delay ($\tau_{d'}$) severely impacts performance, reversing the Nest Score. These results highlight the critical roles of coupling delay, neural adaptation, and integration time window in enabling coherent hierarchical representations. See A.4.3 for further details.

Together, these ablation studies highlight the critical roles of DAE denoising quality and bio-inspired timescale constraints in enabling coherent hierarchical structures in Composer.

7 Related work

7.1 Symbolic and statistical approaches

The problem of parsing visual scenes into part-whole hierarchies has a long history in computer vision and cognitive science [1, 38, 39], where it has been largely addressed with symbolic approaches, from early hand-crafted grammars [39–41] to more recent data-driven structure learning [42, 43]. By representing structural units as explicit discrete symbols, these approaches largely sidestep the representation problem and focus instead on parsing and grammar learning. Here, we consider the problem under distributed representations, which is relevant for understanding how the brain parses visual scenes and essential for realizing parsing in neural networks, yet makes the representation of structure itself non-trivial [4, 5, 44, 45].

7.2 Neural network approaches

Within the neural network paradigm, a closely related and more extensively studied concept is object-centric representation (OCR) [6], which decomposes a scene into a flat set of object representations that can be processed and recombined independently. A part-whole hierarchy instead represents a scene as a multi-level structure, in which the units and the part-whole relations linking them must be inferred jointly and kept consistent across levels [4], making the problem substantially more challenging [4, 44]. Despite this difference, both hinge on the same step: grouping distributed features into structured units as the basis of compositional representation. Approaches to this step thus apply to both and follow two broad strategies: slot-based models assign each unit a dedicated representational slot, whereas coherence-based models let units emerge as coherent groups within a shared representation.

Slot-based models dominate OCR, where feature space is pre-divided into discrete slots, analogous to fixed memory registers in digital computers [6, 7, 9–11]. While conceptually simple and empirically effective, slots are discrete and typically manually pre-defined to match task demands in a content-independent manner—a design choice noted to constrain structural flexibility and hinder generalization to diverse or novel structures [13]. Moreover, how the brain realizes slot boundaries or weight sharing across slots remains unclear [6], posing a challenge to biologically aligned modeling.

Coherence-based models, by contrast, impose soft constraints in network architecture or dynamics so that structural units emerge continuously, inspired by neural mechanisms. Coherence can be defined via complex phase synchrony [12], spike timing alignment [14], or feature rotation [13]. However, both slot- and coherence-based models largely focus on single-level grouping, neglecting the hierarchical structure that underlies part-whole compositionality.

Part-whole hierarchy. Building on spike-timing coherence [14], our work lets both grouping and hierarchical relations emerge continuously through a dynamic process. In contrast, most part-whole models, whether based on graph networks [46], generative models [47], attention [16], or capsules [18], rely on slot-like structural units whose number or type is predefined.

Among coherence-based alternatives, the "islands of identical vectors" framework [4, 17] represents part-whole relations through vector-orientation coherence, but our quantitative evaluation suggests that it does not reliably yield interpretable part-whole structure, a gap likely obscured by prior evaluations based on classification accuracy or limited visualizations [16–18, 46].

7.3 Related mathematical theories

Heteroclinic cycles are closed chains of saddle points linked by heteroclinic orbits: trajectories leave each saddle point along its unstable direction and approach the next, producing a reproducible sequence of transient states [48, 49]. From the perspective of cognitive theory, such dynamics have been hypothesized to support feature binding and hierarchical chunking [50–52]. Our model shares their underlying principle of winnerless competition [48], suggesting that this class of dynamics may also be exploited in artificial neural networks. The dynamical object, however, differs: near an idealized attracting heteroclinic cycle, the dwell time at each saddle point grows without bound and switching slows down asymptotically, whereas our model behaves more like a stable limit cycle, switching between assemblies within a finite number of time steps.

8 Conclusion & Discussion

In this paper, we explore how to represent part-whole hierarchy continuously through neuronal coherence. To this end, we merge insights from neuroscience and machine learning to develop the cortex-inspired Composer, which softly constrains network dynamics so that compositional structure emerges as an attractor state, with distributed features temporally correlated by the self-consistency of content and structure. Bottom-up integration and top-down attention are essential to this process.

As a forward-looking perspective, we suggest that continualizing compositional structure—analogous to the shift from symbolic to connectionist models for content [15]—may be key to learning and generalization. While distributed representations improved the learnability of conceptual content, structural representations remain largely symbolic, relying on discrete partitioning and pre-defined units. This symbolic paradigm limits the network's ability to adapt its structural assumptions through experience and to generalize across structural variations. We argue that structural continuity is equally essential for enabling flexible, experience-driven organization of part-whole hierarchies.

Acknowledgments

This work was supported by the Brain Science and Brain-like Intelligence Technology—National Science and Technology Major Project under Grant 2021ZD0200300. The authors declare no competing interests.

References

[1] Geoffrey E. Hinton. Some demonstrations of the effects of structural descriptions in mental imagery. *Cogn. Sci.*, 3:231–250, 1979.

[2] Daniel Kahneman, Anne Treisman, and Brian J. Gibbs. The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, 24:175–219, 1992.

[3] Peter Thompson. Margaret Thatcher: A new illusion. *Perception*, 9(4):483–484, 1980.

[4] Geoffrey E. Hinton. How to represent part-whole hierarchies in a neural network. *Neural Computation*, 35:413–452, 2023.

[5] Geoffrey E Hinton. Representing part-whole hierarchies in connectionist networks. In *10th Annual Conference Cognitive Science Society Pod*, pages 48–54. Psychology Press, 2019.

[6] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

[7] Klaus Greff, Rupesh Kumar Srivastava, and Jürgen Schmidhuber. Binding via reconstruction clustering. *arXiv preprint arXiv:1511.06418*, 2015.

[8] Klaus Greff, Antti Rasmus, Mathias Berglund, Tele Hao, Harri Valpola, and Jürgen Schmidhuber. Tagger: Deep unsupervised perceptual grouping. *Advances in Neural Information Processing Systems*, 29, 2016.

[9] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. *Advances in Neural Information Processing Systems*, 30, 2017.

[10] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433. PMLR, 2019.

[11] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

[12] Sindy Löwe, Phillip Lippe, Maja Rudolph, and Max Welling. Complex-valued autoencoders for object discovery. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=1PfcmFTXoa`.

[13] Sindy Löwe, Phillip Lippe, Francesco Locatello, and Max Welling. Rotating features for object discovery. *Advances in Neural Information Processing Systems*, 36:59606–59635, 2023.

[14] Hao Zheng, Hui Lin, Rong Zhao, and Luping Shi. Dance of SNN and ANN: Solving binding problem by combining spike timing and reconstructive attention. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 31430–31443. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/cba76ef96c4cd625631ab4d33285b045-Paper-Conference.pdf`.

[15] Timothy T Rogers and James L McClelland. Parallel distributed processing at 25: Further explorations in the microstructure of cognition. *Cognitive science*, 38(6):1024–1077, 2014.

[16] Shuyang Sun, Xiaoyu Yue, Song Bai, and Philip H. S. Torr. Visual parser: Representing part-whole hierarchies with transformers. *ArXiv*, abs/2107.05790, 2021.

[17] Nicola Garau, Niccoló Bisagno, Zeno Sambugaro, and Nicola Conci. Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13679–13688, 2022.

[18] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with EM routing. In *International conference on learning representations*, 2018.

[19] György Buzsáki. Neural syntax: Cell assemblies, synapsembles, and readers. *Neuron*, 68:362–385, 2010.

[20] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[21] György Buzsáki and Andreas Draguhn. Neuronal oscillations in cortical networks. *Science*, 304:1926 – 1929, 2004.

[22] Wolf Singer. Distributed processing and temporal codes in neuronal networks. *Cognitive Neurodynamics*, 3:189 – 196, 2009.

[23] Christoph von der Malsburg. The correlation theory of brain function. In *Models of neural networks*, pages 95–119. Springer, 1994.

[24] Michael N Shadlen and J Anthony Movshon. Synchrony unbound: a critical evaluation of the temporal binding hypothesis. *Neuron*, 24(1):67–77, 1999.

[25] Jeremy M. Wolfe. Forty years after feature integration theory: An introduction to the special issue in honor of the contributions of Anne Treisman. *Attention, Perception, & Psychophysics*, 82:1–6, 2020.

[26] Pieter R. Roelfsema. Solving the binding problem: Assemblies form when neurons enhance their firing rate—they don't need to oscillate or synchronize. *Neuron*, 111:1003–1019, 2023.

[27] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[28] Dmitry Krotov. A new frontier for Hopfield networks. *Nature Reviews Physics*, 5:366–367, 2023.

[29] Keyvan Mahjoory, Jan-Mathijs Schoffelen, Anne Keitel, and Joachim Gross. The frequency gradient of human resting-state brain oscillations follows cortical hierarchies. *eLife*, 9, 2020.

[30] S Murray Sherman and RW Guillery. On the actions that one nerve cell can have on another: distinguishing "drivers" from "modulators". *Proceedings of the National Academy of Sciences*, 95(12):7121–7126, 1998.

[31] Peter König, Andreas K Engel, and Wolf Singer. Integrator or coincidence detector? The role of the cortical neuron revisited. *Trends in neurosciences*, 19(4):130–137, 1996.

[32] György Buzsáki. The brain from inside out. Oxford University Press, USA, 2019.

[33] Jeremy M. Wolfe. Guided search 6.0: An updated model of visual search. *Psychonomic Bulletin & Review*, 28:1060 – 1092, 2021.

[34] T. S. Lee and Mai Lin Nguyen. Dynamics of subjective contour formation in the early visual cortex. *Proceedings of the National Academy of Sciences of the United States of America*, 98 4:1907–11, 2001.

[35] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[36] Jonathan D Victor and Keith P Purpura. Nature and precision of temporal coding in visual cortex: a metric-space analysis. *Journal of neurophysiology*, 76(2):1310–1326, 1996.

[37] Andreas Karl Engel, Pascal Fries, and Wolf Singer. Dynamic predictions: Oscillations and synchrony in top–down processing. *Nature Reviews Neuroscience*, 2:704–716, 2001.

[38] Irving Biederman. Recognition-by-components: a theory of human image understanding. *Psychological review*, 94(2):115, 1987.

[39] King Sun Fu, John E Albus, and Robert H Anderson. *Syntactic pattern recognition and applications*, volume 4. Springer, 1982.

[40] Zhuowen Tu, Xiangrong Chen, Alan L Yuille, and Song-Chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of computer vision*, 63(2):113–140, 2005.

[41] Song-Chun Zhu and David Mumford. A stochastic grammar of images. *Foundations and trends in computer graphics and vision*, 2(4):259–362, 2007.

[42] Shuchen Wu, Noémi Élteto, Ishita Dasgupta, and Eric Schulz. Learning structure from the ground up—hierarchical representation learning by chunking. *Advances in Neural Information Processing Systems*, 35:36706–36721, 2022.

[43] Shuchen Wu, Mirko Thalmann, Peter Dayan, Zeynep Akata, and Eric Schulz. Building, reusing, and generalizing abstract representations from concrete sequences. In *International Conference on Learning Representations*, volume 2025, pages 91473–91496, 2025.

[44] Paul Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, 46(1-2):159–216, 1990.

[45] Geoffrey E Hinton. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46(1-2):47–75, 1990.

[46] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision GNN: An image is worth graph of nodes. *Advances in neural information processing systems*, 35:8291–8303, 2022.

[47] Fei Deng, Zhu Zhi, Donghun Lee, and Sungjin Ahn. Generative scene graph networks. In *International Conference on Learning Representations*, 2021.

[48] M Rabinovich, A Volkovskii, P Lecanda, Ramón Huerta, Henry DI Abarbanel, and Gilles Laurent. Dynamical encoding by networks of competing neuron groups: winnerless competition. *Physical review letters*, 87(6):068102, 2001.

[49] Misha Rabinovich, Ramon Huerta, and Gilles Laurent. Transient dynamics for neural processing. *Science*, 321(5885):48–50, 2008.

[50] Mikhail I Rabinovich, Pablo Varona, Irma Tristan, and Valentin S Afraimovich. Chunking dynamics: heteroclinics in mind. *Frontiers in computational neuroscience*, 8:22, 2014.

[51] Valentin Afraimovich, Xue Gong, and Mikhail Rabinovich. Sequential memory: Binding dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(10), 2015.

[52] Mikhail I Rabinovich and Pablo Varona. Discrete sequential information coding: heteroclinic cognitive dynamics. *Frontiers in Computational Neuroscience*, 12:73, 2018.

[53] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[54] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[55] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[56] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=b9tUk-f_aG`.

[57] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=a68SUt6zFt`. Featured Certification.

[58] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *2023 IEEE/CVF international conference on computer vision (ICCV)*, pages 3992–4003. IEEE, 2023.

[59] Ares Fisher and Rajesh P. N. Rao. Recursive neural programs: Variational learning of image grammars and part-whole hierarchies. *ArXiv*, abs/2206.08462, 2022.

[60] Mikail Khona, Sarthak Chandra, and Ila Fiete. Global modules robustly emerge from local interactions and smooth gradients. *Nature*, pages 1–10, 2025.

[61] Hao Zheng, Hui Lin, and Rong Zhao. GUST: combinatorial generalization by unsupervised grouping with neuronal coherence. *Advances in Neural Information Processing Systems*, 36:32913–32925, 2023.

[62] György Buzsáki, Adrien Peyrache, and John L. Kubie. Emergence of cognition from action. *Cold Spring Harbor symposia on quantitative biology*, 79:41–50, 2014.

[63] Simon B. Eickhoff, R. Todd Constable, and B. T. Thomas Yeo. Topographic organization of the cerebral cortex and brain cartography. *NeuroImage*, 170:332–347, 2017.

[64] Eugene M. Izhikevich. Polychronization: Computation with spikes. *Neural Computation*, 18:245–282, 2006.

[65] Wolf Singer. The cerebral cortex: A delay-coupled recurrent oscillator network? In *Reservoir Computing*, 2021.

[66] Catherine Tallon-Baudry and Olivier Bertrand. Oscillatory gamma activity in humans and its role in object representation. *Trends in cognitive sciences*, 3(4):151–162, 1999.

[67] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12, 2018.

[68] Rodney J. Douglas and Kevan A. C. Martin. Neuronal circuits of the neocortex. *Annual review of neuroscience*, 27:419–51, 2004.

[69] Jonathan C Horton and Daniel L Adams. The cortical column: a structure without a function. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 360(1456):837, 2005.

[70] Nikola T. Markov, Julien Vezoli, Pascal J. P. Chameau, Arnaud Y. Falchier, René Quilodran, Cyril Huissoud, Camille Lamy, Pierre Misery, Pascale Giroud, Shimon Ullman, Pascal Barone, Colette Dehay, Kenneth Knoblauch, and Henry Kennedy. Anatomy of hierarchy: Feedforward and feedback pathways in macaque visual cortex. *The Journal of Comparative Neurology*, 522:225 – 259, 2013.

[71] Brian A Wandell, Serge O Dumoulin, and Alyssa A Brewer. Visual field maps in human cortex. *Neuron*, 56(2):366–383, 2007.

[72] Zejin Lu, Adrien Doerig, Victoria Bosch, Bas Krahmer, Daniel Kaiser, Radoslaw M Cichy, and Tim C Kietzmann. End-to-end topographic networks as models of cortical map formation and human visual behaviour. *Nature Human Behaviour*, 9(9):1975–1991, 2025.

[73] Nelson Spruston. Pyramidal neurons: dendritic structure and synaptic integration. *Nature Reviews Neuroscience*, 9(3):206–221, 2008.

[74] Matthew E Larkum, J Julius Zhu, and Bert Sakmann. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature*, 398(6725):338–341, 1999.

[75] Matthew Larkum. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in neurosciences*, 36(3):141–151, 2013.

[76] Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski. Neuronal dynamics: From single neurons to networks and models of cognition. 2014.

[77] Leo Kozachkov, Jean-Jacques Slotine, and Dmitry Krotov. Neuron–astrocyte associative memory. *Proceedings of the National Academy of Sciences*, 122(21):e2417788122, 2025.

[78] Luca Mazzucato, Alfredo Fontanini, and Giancarlo La Camera. Dynamics of multistable states during ongoing and evoked cortical activity. *The Journal of Neuroscience*, 35:8214 – 8231, 2015.

[79] Giancarlo La Camera, Alfredo Fontanini, and Luca Mazzucato. Cortical computations via metastable activity. *Current opinion in neurobiology*, 58:37–45, 2019.

[80] John D Murray, Alberto Bernacchia, David J Freedman, Ranulfo Romo, Jonathan D Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, et al. A hierarchy of intrinsic timescales across primate cortex. *Nature neuroscience*, 17(12):1661–1663, 2014.

[81] Rishidev Chaudhuri, Kenneth Knoblauch, Marie-Alice Gariel, Henry Kennedy, and Xiao-Jing Wang. A large-scale circuit mechanism for hierarchical dynamical processing in the primate cortex. *Neuron*, 88(2): 419–431, 2015.

[82] George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

[83] Nelson Cowan. The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *Behavioral & Brain Sciences*, 24(1):87, 2001.

[84] Moshe Abeles. Role of the cortical neuron: integrator or coincidence detector? *Israel journal of medical sciences*, 18 1:83–92, 1982.

[85] Daoyun Ji and Matthew A Wilson. Coordinated memory replay in the visual cortex and hippocampus during sleep. *Nature neuroscience*, 10(1):100–107, 2007.

[86] Shengjin Xu, Wanchen Jiang, Mu-ming Poo, and Yang Dan. Activity recall in a visual cortical ensemble. *Nature neuroscience*, 15(3):449–455, 2012.

[87] Jeffrey P Gavornik and Mark F Bear. Learned spatiotemporal sequence recognition and prediction in primary visual cortex. *Nature neuroscience*, 17(5):732–737, 2014.

[88] Georg Northoff and Zirui Huang. How do the brain's time and space mediate consciousness and its different dimensions? temporo-spatial theory of consciousness (ttc). *Neuroscience & Biobehavioral Reviews*, 80:630–645, 2017.

[89] Shiming Tang, Yimeng Zhang, Zhihao Li, Ming Li, Fang Liu, Hongfei Jiang, and Tai Sing Lee. Large-scale two-photon imaging revealed super-sparse population codes in the V1 superficial layer of awake monkeys. *eLife*, 7, 2018.

[90] Xiaoqin Wang. Cortical coding of auditory features. *Annual review of neuroscience*, 41:527–552, 2018.

[91] Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2:79–87, 1999.

[92] André Moraes Bastos, W. Martin Usrey, Rick A Adams, George R. Mangun, Pascal Fries, and Karl J. Friston. Canonical microcircuits for predictive coding. *Neuron*, 76:695–711, 2012.

[93] Karl J. Friston. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11: 127–138, 2010.

[94] Maria K. Eckstein, Christopher Summerfield, Nathaniel D. Daw, and Kevin J. Miller. Predictive and interpretable: Combining artificial neural networks and classic cognitive models to understand human learning and decision making. *bioRxiv*, 2023.

[95] Joshua C. Peterson, David D. Bourgin, Mayank Agrawal, Daniel Reichman, and Thomas L. Griffiths. Using large-scale experiments and machine learning to discover theories of human decision-making. *Science*, 372:1209 – 1214, 2021.

[96] Jan Melchior, Robin Schiewer, and Laurenz Wiskott. Hebbian descent: A unified view on log-likelihood learning. *Neural Computation*, 36(9):1669–1712, 2024. doi: 10.1162/neco_a_01684.

[97] Wolf Singer. Neuronal synchrony: a versatile code for the definition of relations? *Neuron*, 24(1):49–65, 1999.

[98] Andreas Karl Engel and Wolf Singer. Temporal binding and the neural correlates of sensory awareness. *Trends in Cognitive Sciences*, 5:16–25, 2001.

[99] Alexander Thiele and Gene Stoner. Neuronal synchrony does not correlate with motion coherence in cortical area MT. *Nature*, 421(6921):366–370, 2003.

[100] Pieter R Roelfsema, Victor AF Lamme, and Henk Spekreijse. Synchrony and covariation of firing rates in the primary visual cortex during contour grouping. *Nature neuroscience*, 7(9):982–991, 2004.

[101] Pieter R Roelfsema. Cortical algorithms for perceptual grouping. *Annu. Rev. Neurosci.*, 29(1):203–227, 2006.

[102] Mohamady El-Gaby, Adam Loyd Harris, James CR Whittington, William Dorrell, Arya Bhomick, Mark E Walton, Thomas Akam, and Timothy EJ Behrens. A cellular basis for mapping behavioural structure. *Nature*, 636(8043):671–680, 2024.

[103] Lars Buesing, Johannes Bill, Bernhard Nessler, and Wolfgang Maass. Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS computational biology*, 7(11):e1002211, 2011.

[104] Christoph von der Malsburg. The what and why of binding: the modeler's perspective. *Neuron*, 24(1): 95–104, 1999.

[105] Sunny Nigam, Sorin Pojoga, and Valentin Dragoi. A distinct population of heterogeneously color-tuned neurons in macaque visual cortex. *Science Advances*, 7(8):eabc5837, 2021.

[106] Thomas Wachtler, Terrence J Sejnowski, and Thomas D Albright. Representation of color stimuli in awake macaque primary visual cortex. *Neuron*, 37(4):681–691, 2003.

[107] Bevil R Conway, Sebastian Moeller, and Doris Y Tsao. Specialized color modules in macaque extrastriate cortex. *Neuron*, 56(3):560–573, 2007.

[108] Kaitlin S Bohon, Katherine L Hermann, Thorsten Hansen, and Bevil R Conway. Representation of perceptual color space in macaque posterior inferior temporal cortex (the V4 complex). *eNeuro*, 3(4), 2016.

[109] Jing Pei, Lei Deng, Sen Song, Mingguo Zhao, Youhui Zhang, Shuang Wu, Guanrui Wang, Zhe Zou, Zhenzhi Wu, Wei He, Feng Chen, Ning Deng, Si Wu, Yu Wang, Yujie Wu, Zheyu Yang, Cheng Ma, Guoqi Li, Wentao Han, Huanglong Li, Huaqiang Wu, Rong Zhao, Yuan Xie, and Luping Shi. Towards artificial general intelligence with hybrid Tianjic chip architecture. *Nature*, 572:106 – 111, 2019.

Contents

A Appendix **18**

A.1 Limitations and Future Directions 19

A.1.1 Use of Synthetic Data and Generalization 19

A.1.2 Simplified Architecture 19

A.1.3 Simplified Representation Hypothesis 20

A.1.4 Simplified Computation Across Levels 21

A.1.5 Learning Scheme and Priors 21

A.1.6 Scope of Evaluation Metrics and Baselines 21

A.2 Code availability 22

A.3 Metric 23

A.3.1 Spike Train Distance: Victor-Purpura Metric 23

A.3.2 Cluster Coherence: Silhouette Score 23

A.3.3 Silhouette Score Applied to Neuronal Coherence 24

A.3.4 Scores to Measure the Part-Whole Hierarchy 24

A.3.5 Perturbation Study 26

A.3.6 The shift: from synchronization to polychronization 27

A.3.7 Generalization to Other Models 27

A.4 How does the Composer Work? 33

A.4.1 Initialization of the Dynamics 33

A.4.2 Time scale parameters 33

A.4.3 Ablation Study of Timescale Parameters 34

A.4.4 Parameter Sensitivity Test 35

A.5 Implementation of Training 37

A.5.1 Resources 37

A.5.2 Network architecture and training hyperparameters 37

A.5.3 Dataset for Training the DAE 37

A.5.4 Loss Function 38

A.5.5 Ablation study of the DAE 38

A.6 Relation to Neuroscience 39

A.6.1 Circuit architecture. 39

A.6.2 Dynamics. 40

A.6.3 Representation. 41

A.6.4 Learning and priors. 42

A.6.5 Open debates and alternative views. 42

A.7 Benchmarking 43

A.7.1 Composer scores (corresponding to Fig.9) 43

A.7.2 Agglomerator scores (corresponding to Fig.9) 43

A.7.3 Introducing the Agglomerator 43

A.7.4 How is coherence measured in the Agglomerator? 44
A.7.5 The architecture detail of the Agglomerator 45
A.7.6 Down-scaling of the ground truth 45
A.7.7 Limitations of the Agglomerator under structured evaluation 46
A.8 Additional experimental details and results 48
A.8.1 Convergence behavior across datasets 48
A.8.2 Varying the Number of Objects Across Datasets 48
A.8.3 Visualization 48
A.8.4 Additional visualizations 49
A.8.5 Multi-stability in part-whole inference 49
A.8.6 Three-level hierarchy 50
A.8.7 Encoding RGB images 50

A Appendix

Appendix Overview

- **General Discussions** (Sec. A.1–A.2): Limitations (Sec.A.1), future directions (Sec.A.1), and code availability (Sec.A.2).
- **Metrics** (Sec. A.3): Step-by-step description of how structure-aware metrics are developed, and how they generalize to broader settings.
- **Model Details and Dynamics** (Sec. A.4–A.6): Initialization (Sec. A.4.1), time scale configuration and robustness (Sec. A.4.2, A.4.3–A.4.4), training of DAE (Sec. A.5, A.5.5), and biological motivation (Sec. A.6).
- **Benchmark Model** (Sec. A.7): Introduction of the benchmark model and application of evaluation metrics to the non-spiking setting.
- **Experiment Details** (Sec. A.8.1, Sec. A.8.2, Sec. A.8.3): Clarifications and settings for main text experiments.
- **Additional Visualizations** (Sec. A.8.4): More qualitative results across datasets, including failure cases.
- **Additional Results** (Sec. A.8.5, A.8.6, A.8.7): (1) Multi-stable inference on ambiguous inputs; (2) Extension to 3-level hierarchies; (3) Extension to RGB images.

A.1 Limitations and Future Directions

We group below several key limitations and possible extensions to the current framework. Each subsection highlights an aspect of the design that could be extended or generalized in future work, while maintaining compatibility with the current Composer architecture.

A.1.1 Use of Synthetic Data and Generalization

Current setting. Evaluation is performed on synthetic images with known part-whole structure to enable explicit validation. Compared with MNIST [53] or real-world classification datasets such as CIFAR [54] and ImageNet [55], where each image is typically centered on a single object and labeled at a single level, our datasets are structurally richer: each scene contains multiple objects, each composed of multiple parts, with annotations at both levels. Being synthetically generated, however, they lack the feature complexity of real images.

Limitation. Synthetic datasets provide controlled conditions but lack real-world complexity such as pattern diversity, feature richness, diverse relative pose between parts and wholes, 3D objects, occlusion, clutter, and noise. The current results therefore demonstrate feasibility in a controlled setting, and the method has yet to be evaluated on natural images.

Future direction. While synthetic evaluation offers a well-controlled starting point, extending the approach to real-world images remains an important direction for future work. This requires models to handle greater variability and ambiguity, and may motivate the development of new benchmarks with annotated part-whole hierarchies.

Recent studies [13, 56] provide insightful and practical strategies for extending object-centric models originally developed for synthetic datasets to real-world images. By augmenting these models with powerful pretrained encoders such as DINO [57], high-dimensional inputs are transformed into compact latent representations that exhibit structural regularities comparable to those in synthetic data, enabling successful scaling of single-level OCR models to natural scenes [13, 56].

Datasets like SA-1B from the Segment Anything project [58] offer additional promise. Some samples include multi-level segmentations and reflect annotation ambiguity. However, SA-1B remains agnostic to explicit part-whole relationships, limiting its immediate suitability. Integrating large language model (LLM) prompts to generate nested masks may provide a path forward.

Finally, we emphasize that this scaling process should be approached with caution, ensuring that improvements in task performance do not come at the expense of representational structure or interpretability. Evaluations should prioritize structure-aware analysis rather than task-level metrics such as classification accuracy, which are agnostic to internal organization.

A.1.2 Simplified Architecture

A.1.2.1 Discrete Separation of Levels

Current setting. While grouping within each level and cross-level nestedness are continuously represented, the model imposes one discrete constraint: the representation space is explicitly separated into distinct levels. The existence and number of levels are therefore set by design rather than emergent. Relatedly, the whole level is assigned a slower timescale than the part level, and what each level represents is determined by its pretrained DAE (Sec. A.1.5).

Limitation. While this design is consistent with GLOM [4] and other part-whole models [16, 17, 59], it reflects a broader modeling choice that may introduce general limitations: (1) the number of levels cannot be dynamically inferred from content; (2) the maximum number of levels is fixed by architectural design; and (3) the brain may not exhibit such sharply defined boundaries, though the extent to which this separation aligns with biological organization remains an open question.

Clarification and Future direction. Despite these concerns, the number of levels humans actively process at any moment is thought to be limited—possibly no more than five [4]. Thus, a fixed small number of discrete levels may suffice for most visual scenes. Moreover, the same model architecture can be reused across vastly different hierarchies (e.g., molecules vs. galaxies) [4], supporting the design choice of discrete, bounded levels.

More fundamentally, it remains an open question how modular structure emerges in the brain. Recent findings provide a compelling insight: discrete modular organization can emerge from continuous spatial or structural gradients, as observed in biological systems [60]. In addition, many biophysical parameters in the brain follow long-tailed or skewed distributions, which may serve as distributed, continuous constraints underlying hierarchical organization. Future work could explore whether such structure can emerge in dynamical recurrent networks by inducing heterogeneity in timescales, delays, synaptic strengths, or connectivity patterns.

A.1.2.2 Minimal Realization of DAE

Current setting. In this work, the trainable DAE module is implemented as a minimal, lightweight MLP to isolate the core mechanism without introducing unnecessary complexity.

Limitation. As shown in the main text, parsing quality correlates with the denoising strength of the DAE (see also Sec. A.1.5 for the role of the priors it encodes). This minimal design limits the model's ability to handle more complex patterns. However, this is a technical choice rather than a fundamental architectural constraint.

Future direction. Numerous advanced variants of denoising autoencoders (DAE), such as diffusion models, offer greater expressive capacity and can better denoise complex inputs. Future work may incorporate stronger DAE architectures to improve denoising performance and, in turn, enhance the model's ability to support robust structure emergence on more challenging dataset.

A.1.3 Simplified Representation Hypothesis

A.1.3.1 Two Hierarchical Levels

Current setting and limitation. For simplicity, this work focuses on two levels—part and whole—leaving open the model's ability to handle deeper hierarchies, which are common in vision and cognition. The maximum number of levels is also fixed by design (see A.1.2.1).

Future direction. It has been suggested that only a limited number of levels (e.g., no more than five) are needed to support human perception [4], making the two-level case representative for general analysis. Importantly, the part and whole levels in our model share the same representational format: both are spatially aligned with the input, group features through the same coherence mechanism, and each holds a complete feature space for its level. The whole level can therefore serve as the part level of a further level above it, so the two-level structure forms a building block that can be reused to build deeper hierarchies. This is not the case in prior work such as [16], where the whole level consists of spatial patches whereas the part level consists of slots without spatial layout, so the same structure cannot be reused at the next level. All components of the Composer are compatible with such extension, and Fig. 31 demonstrates its application to a three-level structure.

A.1.3.2 Feature as Pixel Presence

Current setting. The current model encodes part-whole structure in a pixel-wise spike coding space (SCS), topographically aligned with the input image. This one-to-one spatial mapping facilitates interpretability and supports direct visualization of spiking activity, attention signals ($\gamma_1, \gamma_2, \Gamma$), and part-whole labels ($s_1, s_2, \mathrm{GT}_1, \mathrm{GT}_2$), as shown in Fig. 6 and Fig. 7.

Limitation. Features are currently defined as binary pixel presence, and objects as spatial groupings of these pixels. Although the current implementation uses pixel-wise spiking neurons, each neuron can be abstractly interpreted as being selective to the presence of a particular feature at its location, and objects as compositions of such feature-selective activations. Under this view, the mechanism does not depend on features being pixels: pixel presence is simply the most basic case of feature presence, so the model is compatible with general feature spaces beyond raw pixels. However, in the current implementation, each location is represented by a single binary spiking neuron, which limits the ability to encode complex attributes such as color, texture, or shape.

Future direction. A natural extension is to assign a population of spiking neurons to each spatial location to encode richer local features, analogous to capsule networks [18] or cortical mini-columns [4]. For instance, each neuron could be selective to specific feature values (e.g., RGB components). A preliminary implementation is provided in Sec. A.8.7 and Fig. 32. With such richer features,

the identity mapping between levels would also need to be generalized (Sec. A.1.4). Beyond this, part-whole representations could be lifted into the latent space of the DAE (as in Slot Attention [11]), enabling more abstract encoding and flexible cross-level interaction. As discussed in A.1.1, model inputs can also be generalized to tokenized embeddings from pretrained encoders (e.g., DINO [57]), allowing feature neurons to operate over higher-level representations and potentially improving scalability to real-world data.

A.1.4 Simplified Computation Across Levels

Current setting. As in GLOM [4], part and whole representations are spatially aligned, so that each location at one level corresponds to the same location at the other. Here a location is a pixel, but it can more generally be a patch or a token. At each location, the mapping between levels is simplified to an identity mapping: for example, part-level activity is integrated over time and passed directly to the whole level. This simplification suffices because features are currently defined as binary pixel presence (Sec. A.1.3.2), so both levels share the same feature space at each location.

Limitation. Spatial alignment itself is a common design choice in part-whole models. For richer feature spaces, however, such as when each location is a patch encoding local color or texture, an identity mapping is no longer sufficient: it ignores differences in pose between parts and wholes, as well as differences in the features represented at each level.

Future direction. Generalizing the identity mapping is natural within the current framework. As in GLOM, where the bottom-up and top-down predictions at each location are computed by neural networks [4], the identity mapping at each location could be replaced with a neural network module that transforms activity between levels, for example to account for pose transformations between parts and wholes.

A.1.5 Learning Scheme and Priors

Current setting. Each DAE is pretrained on objects of its own level and thus encodes level-specific semantic priors: the part-level DAE encodes the appearance of parts, and the whole-level DAE that of wholes. These priors can be interpreted as evolutionarily acquired configurations or learned object prototypes. Pretraining involves no loss that compares the representation with a target structure; how a given scene is grouped into parts and wholes, and how these are nested, is inferred by the dynamics of the Composer.

Limitation. These level-specific priors are central to the model's behavior: parsing quality correlates with the denoising quality of the DAEs (Fig. 10; see also Fig. 17 and Fig. 20). The contributions of these priors and of the inference dynamics to the emergent hierarchy have therefore not been fully disentangled. Moreover, the framework assumes that object prototypes are available in advance, leaving open how part-whole structure can emerge when they must also be learned.

Future direction. Recent work [61] demonstrates that predictive coding objectives alone can drive the unsupervised emergence of cell assemblies, even without pretrained content priors. This is achieved by replacing pretraining with a compact end-to-end predictive loss on the same DAE architecture. This insight opens a path toward generalizing the current model into an end-to-end unsupervised framework. Extending the Composer in this direction may enable fully emergent structure learning, consistent with neuroscientific theories such as "Cognition from Action" [62].

More broadly, the Composer parameterizes structural constraints using distributed and continuous variables (e.g., weights, timescales), which provides a natural inductive bias for gradual structure learning—an advantage over discrete slot-based approaches. The potential of this formulation has been demonstrated in [61] for the single-level case; extending this architectural principle to support hierarchical structure learning represents a promising direction for future work.

A.1.6 Scope of Evaluation Metrics and Baselines

Current setting. The Part, Whole, and Nest Scores evaluate grouping and cross-level relations defined through coherence, i.e., similarity among unit activities. They can therefore in principle be applied to models that define grouping in this way.

Limitation. The metrics assume that grouping is defined through coherence and, for the Nest Score, that the two levels are aligned one to one in space, so that ground-truth labels apply across levels.

They are therefore not directly applicable to representations such as slots or tokens. Many such models nonetheless imply a similarity among units, for example between the attention maps of different slots [16], so the metrics can be adapted to their representational assumptions. For the same reason, our quantitative comparison is mainly against Agglomerator, a coherence-based model that can be evaluated directly. Other part-whole models adopt diverse representational formats, such as slots, tokens, graphs, or capsules, and typically neither specify how these representations encode parts, wholes, and the relations between them, nor rigorously evaluate them in the original work; evaluating these models would first require substantially reinterpreting their representations, and a broader comparison is left for future work.

Future direction. Building on the proposed metrics and benchmarks, future work could adapt the metrics to other representational formats, evaluate a wider range of part-whole models, and construct natural-image datasets with multi-level part-whole annotations, moving toward a general evaluation framework for part-whole representations.

In summary, the current results establish the feasibility of representing and inferring part-whole hierarchies through nested neuronal coherence in a controlled setting. Building on this, the framework opens several natural directions for extension, including enhanced local feature encoding, advanced DAE architectures, tokenized inputs, cross-level transformations, deeper hierarchical composition, and unsupervised structure learning. All of these remain architecturally compatible with the Composer and may support broader generalization as well as a closer link to biological computation.

A.2 Code availability

The source code for results in the paper can be found at:

https://github.com/HaoZ24/Representing-Part-Whole-Hierarchy-with-Nested-Neuronal-Coherence.

A.3 Metric

To evaluate neuronal coherence and part-whole organization, we define a family of metrics based on the Silhouette Score applied to spike trains. Conceptually, synchronized cell assemblies are treated as clusters of spike trains, and coherence is quantified as the intra-cluster similarity induced by ground-truth assignments.

This section introduces the full evaluation pipeline. We first introduce a spike train distance metric based on the Victor-Purpura (VP) measure [36], then review the Silhouette Score for clustering coherence (Sec. A.3.2). Together, these define a set of coherence scores—**Part Score**, **Whole Score**, and **Nest Score**—that jointly quantify hierarchical organization. A perturbation-based validation of the scores is presented in Sec. A.3.5, and Sec. A.3.7 discusses generalization to evaluate other coherence-based models such as "identical islands of vectors" [4]. Evaluation hyperparameters are summarized in Table 2.

A.3.1 Spike Train Distance: Victor-Purpura Metric

The Victor-Purpura (VP) metric [36] is a classical non-Euclidean distance designed to compare spike trains by accounting for the timing and structure of discrete events. Unlike Euclidean distances, which require fixed-length vector representations, the VP metric provides a more natural fit for the variable and temporally structured nature of spike trains.

The metric treats spike trains as symbol sequences and defines distance as the minimal cost of transforming one spike train into another using three operations (Fig.11): inserting or deleting a spike (cost = 1), or shifting a spike by Δt (cost = $\Delta t/\tau$), where τ is a time-scale parameter controlling temporal precision. Formally:

$$D_{VP}(s_i, s_j; \tau^{-1}) = \min_{T} \sum_{u=1}^{|T|} \text{cost}(T(u)), \quad T(u) \in \{\text{add}, \text{delete}, \text{shift}_{\Delta t}\}, \tag{11}$$

with $\text{cost}(\text{add}) = \text{cost}(\text{delete}) = 1$ and $\text{cost}(\text{shift}_{\Delta t}) = \Delta t/\tau$.

This formulation satisfies the axioms of a valid metric (positivity, symmetry, triangle inequality), and induces a well-defined metric space over spike trains. The minimal transformation cost can be computed efficiently via dynamic programming.

A key feature of the VP metric is the temporal resolution parameter τ, which modulates sensitivity: as $\tau \to \infty$, shifting incurs no cost and the metric reflects spike count (rate coding); as $\tau \to 0$, only perfectly synchronous spike trains have zero distance, making the metric sensitive to precise timing (temporal coding).

In our evaluation, we assign different time scales for each level: part-level coherence is computed using a smaller τ_p to capture fast local synchrony, while whole-level coherence uses a larger τ_w. The Nest Score, which compares across levels, is also evaluated using τ_w, consistent with the assumption that child assemblies are temporally nested within parent windows.

A.3.2 Cluster Coherence: Silhouette Score

The Silhouette Score [35] is a standard metric for assessing clustering quality. For each sample i, it compares the average intra-cluster distance $a(i)$ to the average nearest-cluster distance $b(i)$, computing:

$$\text{Silhouette}(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}. \tag{12}$$

The final Silhouette Score is computed as the average over all samples. Scores close to 1 indicate that samples are well matched to their assigned clusters and clearly separated from others; values near 0 suggest ambiguous structure or poor separability; and negative values indicate that samples may be closer to a different cluster than their own.

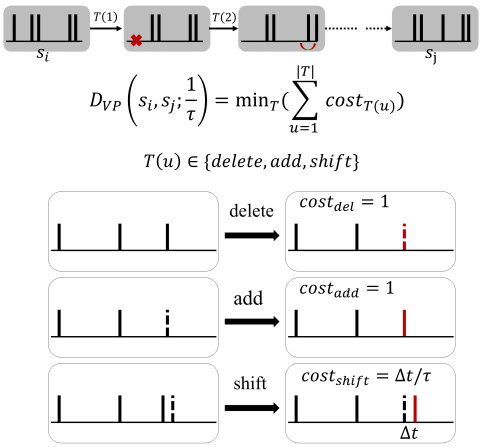

Figure 11: Victor-Purpura metric. Top: An example sequence of operations to transform one spike train s_i to another s_j. Middle: VP distance between s_i and s_j is defined as the minimal total cost to transform one to the other. Bottom: three operations and their respective costs (add/delete = 1; shift = $\Delta t/\tau$). Dashed bars indicate removed spikes or spikes to be added by the operation; red bars highlight the outcome of the transformation.

A.3.3 Silhouette Score Applied to Neuronal Coherence

To measure neuronal coherence, we interpret synchronized cell assemblies as clusters of spike trains. Ground-truth part or whole assignments are used as cluster labels. Each neuron is treated as a sample, its spike train as the feature vector, and the VP metric as the distance function. The Silhouette Score computed on this setup reflects how tightly spike trains within the same cluster synchronize, and how well-separated different assemblies are.

Formally, this VP-induced Silhouette Score captures intra-cluster temporal similarity and inter-cluster separation in spike timing space. High values reflect strong synchrony within assemblies; values near 0 indicate temporally unstructured activity, such as random firing or overlapping assemblies; negative values suggest that neurons are more synchronous with other assemblies than with their assigned group.

Evaluation is performed on a post-convergence segment of the simulation, with duration τ_l chosen to cover at least one oscillation cycle (Table 2).

A.3.4 Scores to Measure the Part-Whole Hierarchy

To quantify part-whole structure, we define three coherence scores using the VP-Silhouette Score, computed over spike trains and ground-truth cluster labels at each level.

Part Score:

$$\text{Part Score} = \text{Silhouette}(VP(spk_1, spk_1; \tau_p), \text{label}_1), \tag{13}$$

where $spk_1 \in \{0,1\}^{N \times \tau_l}$ are spike trains at the part level, VP(spk_1, spk_1) is the pairwise distance matrix, with the (i,j)-th entry giving the Victor-Purpura distance between the i-th and j-th part-level neurons, label_1 denotes the ground-truth part-object assignment, and τ_p controls the temporal sensitivity of the VP metric. This score reflects how well neurons assigned to the same part object synchronize temporally. A score of 1 indicates perfect within-group synchrony; scores near 0 reflect random or weakly structured firing; negative values suggest stronger synchrony with a different group.

Whole Score:

$$\text{Whole Score} = \text{Silhouette}(VP(spk_2, spk_2; \tau_w), \text{label}_2), \tag{14}$$

where $spk_2 \in \{0,1\}^{N \times \tau_l}$ and label_2 denote the whole-level spike trains and their ground-truth object assignments, respectively. $\tau_w > \tau_p$ reflects the slower timescale of whole-object representations.

Part and Whole Scores measure intra-level coherence independently, but do not capture alignment across levels. For instance, cell assemblies can be globally permuted or translated without affecting these scores, even though such transformations may disrupt the underlying hierarchy (Fig. 16).

To address this, we define a third metric:

Nest Score:

$$\text{Nest Score} = \frac{4}{3} \cdot \text{Silhouette}(VP(spk_1, spk_2; \tau_w), \text{label}_2) \tag{15}$$

Here, $spk_1, spk_2 \in \{0,1\}^{N \times \tau_l}$ denote the part- and whole-level spike trains, respectively. $VP(spk_1, spk_2) \in \mathbb{R}^{N \times N}$ is the pairwise VP distance matrix across levels, with entry (i, j) representing the distance between the i-th part-level and j-th whole-level neuron. label_2 provides the cluster labels for whole-level assignments. Intuitively, Nest Score captures how tightly part-level assemblies align with whole-level assemblies in time, assuming nested temporal structure.

Following the Silhouette Score formulation (Eq. 12), we compute the Nest Score as the mean of $\frac{b(i)-a(i)}{\max(a(i),b(i))}$, where distances are computed across levels using $VP(spk_1, spk_2; \tau_w)$. For each whole-level neuron i, $a(i)$ is the average distance to all part-level neurons assigned to the same whole-level object (intra-group; Fig. 12a, green box), and $b(i)$ is the average distance to all part-level neurons in a nearest different group (inter-group; Fig. 12a, blue box). This extends the standard Silhouette formulation by measuring temporal coherence across hierarchical levels.

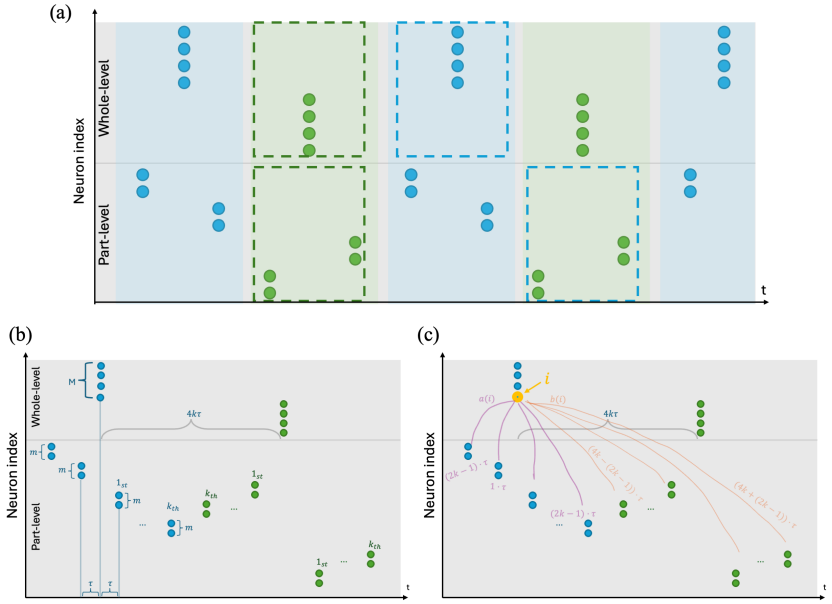

Figure 12: Illustration of the Nest Score. (a) Schematic diagram for cross-level distance: green box shows intra-group, blue box shows inter-group. Each circle denotes a spike; color encodes the whole-level assignment shared by both part- and whole-level neurons. (b, c) Schematic diagrams supporting the proof of the universal normalization factor (Theorem 1). (b) Idealized nested structure illustrating definitions of M, m, and minimal temporal distance τ. (c) Showing how to compute $a(i)$ and $b(i)$ for a selected whole-level neuron (yellow). $2k$ part assemblies are symmetrically arranged around each whole assembly, labeled from 1st to k-th on both sides.

Because part and whole assemblies encode structurally distinct levels, their activities are temporally coordinated but not identical. The part level contains more fine-grained assemblies nested within broader whole-level patterns, precluding perfect cross-level alignment even in ideal cases. As a result, the raw cross-level Silhouette cannot reach 1. We therefore normalize it by its theoretical

maximum—$3/4$—to match the [–1, 1] range of the other scores. The scaling factor $4/3$ brings the ideal nested score to 1, as confirmed empirically (e.g., Fig. 16)[1].

This formulation assumes a one-to-one spatial alignment between part- and whole-level spike coding spaces (SCS), such that neurons at both levels correspond to the same pixel locations. This topographic alignment enables meaningful cross-level comparisons and allows the use of ground-truth labels from one level (e.g., whole-level) to evaluate coherence across levels. The assumption reflects a cortical organization principle often referred to as topographic mapping [63], and is consistent with related modeling work [4]. While this design greatly simplifies evaluation, relaxing the alignment constraint may be an important direction for future extensions.

Lastly, we provide the proof that the normalization factor is universal, not dependent on specific parsing configuration of the hierarchical structure as the theorem below:

Theorem 1. *For any ideal nested hierarchy of assemblies with even branching factor (e.g., binary, quaternary), the unnormalized Nest Score equals $3/4$, independent of the branching number.*

Proof. For each whole-level neuron i, we define the intra-group distance $a(i)$ as the average VP distance between i and all part-level neurons aligned to the same whole-level object, based on whole-level ground-truth cluster labels. Similarly, $b(i)$ denotes the average distance from i to part-level neurons in the nearest other group (Fig.12a). Here we compute $a(i), b(i)$ for an ideal parsing structure.

Let there be K whole-level objects, each with $2k$ part-level components. Each whole object is represented by M neurons, and each part by m (Fig.12b). For perfectly nested structure, assume that cell assemblies at both levels are uniformly distributed along the time axis, and that the minimal temporal distance between aligned part and whole assemblies is τ (Fig.12b). Then we have (Fig.12c) :

$$a(i) = 2 \cdot m \cdot \sum_{j=1}^{k} (2j-1)\tau = m \cdot 2k^2\tau$$

For the inter-group distance (Fig.12c):

$$b(i) = m \cdot \sum_{j=-k+1}^{k} (4k+2j-1)\tau = m \cdot 8k^2\tau$$

Thus,

$$\text{Silhouette}(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} = \frac{6k^2}{8k^2} = \frac{3}{4}$$

Since i was arbitrary, the mean over all neurons is also $3/4$. □

A.3.5 Perturbation Study

To validate the proposed scores, we conduct a perturbation study (Fig. 5) and provide detailed results in Figs. 13–16. The study begins with a manually constructed, perfectly nested spike pattern over two oscillation cycles (total duration $2\tau_{\text{total}} = 54$ time steps), as shown in Fig. 5a. Part- and whole-level assemblies are temporally synchronized and uniformly spaced.

Neuron-level perturbation: independent noise generated for each neuron

We then introduce perturbations by randomly shifting each spike to a nearby time step:

$$t_i \to t_i \pm \Delta t_i, \quad \Delta t_i \leq \tau \tag{16}$$

where t_i is the spike time of neuron i, and τ controls the perturbation scale. Δt_i is independently generated for each neuron. When $\tau = 0$, no perturbation occurs. At $\tau = 0.5 \cdot \tau_{\text{total}}$, spike times become fully randomized. We define the perturbation level as $\tau/(0.5 \cdot \tau_{\text{total}}) \in [0\%, 100\%]$.

In Fig. 5b & Fig. 13, perturbations are applied to both levels. All three scores decrease smoothly from 1 to 0, validating their sensitivity. In Fig. 5c & Fig. 14, only the part level is perturbed. As expected,

[1] It can be analytically proved that this normalization factor is universal: it is not dependent on the specific shape of the hierarchical structure (The number of branches). For example, both quad-tree and binary-tree share the same normalization factors. We provide a proof as Theorem 1 at the end of this section

Part and Nest Scores drop while the Whole Score remains stable. Conversely, in Fig. 5d & Fig. 15, only the whole level is perturbed, and Part Score remains unaffected. These examples confirm that Part and Whole Scores independently reflect intra-level coherence, while stronger grouping at each level also contributes to a higher Nest Score, indicating better cross-level structure.

Group-level perturbation: Correlated noise generated for each group

$$t_i \to t_i \pm \Delta t_k, \quad \Delta t_k \leq \tau, \quad i \in C_k \tag{17}$$

To isolate the contribution of the Nest Score, we construct a perfectly nested configuration (as in Fig. 5a) and perturb entire cell assemblies C_k rather than individual spikes. Specifically, each synchronized group is shifted by a shared offset Δt_k, sampled independently per group (Fig. 5e top & Fig. 16). This preserves intra-group synchrony while disrupting the relative timing between groups.

As shown in Fig. 5e and Fig. 16, the Nest Score decreases smoothly with increasing perturbation, while the Part and Whole Scores remain close to 1. This reflects a breakdown in hierarchical coordination: part-level assemblies may become more aligned with unrelated whole-level assemblies. Notably, the Nest Score can drop below zero, indicating systematic cross-level misalignment.

These perturbation experiments confirm that the proposed metrics effectively distinguish intra-level coherence from cross-level nestedness. They also provide empirical support for the universality of the normalization factor $(4/3)$ under ideal nested conditions.

A.3.6 The shift: from synchronization to polychronization

In biological neural systems, spike timing synchronization plays a critical role in downstream integration, as coincident inputs are more likely to drive postsynaptic responses [19]. However, due to heterogeneous axonal delays, synchronized arrival may result from spikes fired at consistent temporal offsets—a phenomenon known as polychronization [64]. Polychronization generalizes the notion of synchrony by allowing fixed temporal shifts within a group, preserving functional coherence at the readout level.

In our setting, such fixed cross-level shifts are not considered structural deviations. A constant temporal offset is applied to align whole-level spike trains with the part-level before computing the Nest Score. This adjustment accounts for expected polychrony and preserves the intended hierarchical nesting. The shift magnitude is small, fixed, and documented in Table 2.

This compensation also applies to visualizations (e.g., Fig. 6), aligning assemblies without altering interpretation. Overall, this design reflects the idea that perfect synchrony across levels is not required, and fixed offsets can be neurally meaningful.

A.3.7 Generalization to Other Models

The proposed metrics can be extended to evaluate other models that construct part-whole hierarchies based on similarity or coherence. For example, GLOM [4] and its implementation Agglomerator [17] organize representations via vector similarity rather than spike synchrony. In such settings, the Victor-Purpura metric is no longer required. Instead, each vector (e.g., from a model column) is treated as a sample, and its dimensions as features, enabling direct application of the Silhouette Score.

Let l_p and l_w be the feature vectors at the part and whole levels, respectively. Then the coherence scores become:

$$\text{Part Score} = \text{Silhouette}(D(l_p, l_p), \text{label}_1) \tag{18}$$

$$\text{Whole Score} = \text{Silhouette}(D(l_w, l_w), \text{label}_2) \tag{19}$$

$$\text{Nest Score} = \alpha \cdot \text{Silhouette}(D(l_p, l_w), \text{label}_2) \tag{20}$$

Here, D denotes a standard Euclidean distance and α is a normalization constant, e.g., calibrated on idealized structures. For GLOM, l_p and l_w represent vectors for part- and whole-level columns. Notably, the "islands of identical vectors" in GLOM are analogous to correlated assemblies in our model, where temporally structured spike trains form population codes.

Table 2: Evaluation parameters. τ_p and τ_w are the timescale parameters used to compute VP distances for the part and whole levels, respectively. "Shift" denotes the fixed time offset applied for computing the Nest Score and for visualization. We report two segment lengths: τ_l for computing scores, and a longer segment for visualization, chosen to include more oscillatory cycles and better reveal temporal structure.

Dataset	**Ts**	**Squares**	**SHOPs**	**Double MNIST**
τ_p	2	2	3	2
τ_w	6	7	6	4
shift	10	9	6	2
segment length (score,τ_l)	160	75	42	32
segment length (visualization)	200	100	100	70

This formulation also applies to complex-valued networks [12], where similarity is encoded in the phase. The same structure applies with l_p, l_w denoting phase vectors.

In sum, the proposed metrics generalize to coherence-based part-whole models tested on synthetic datasets with available ground truth. Extending them to real-world data without annotations remains a challenge for future work.

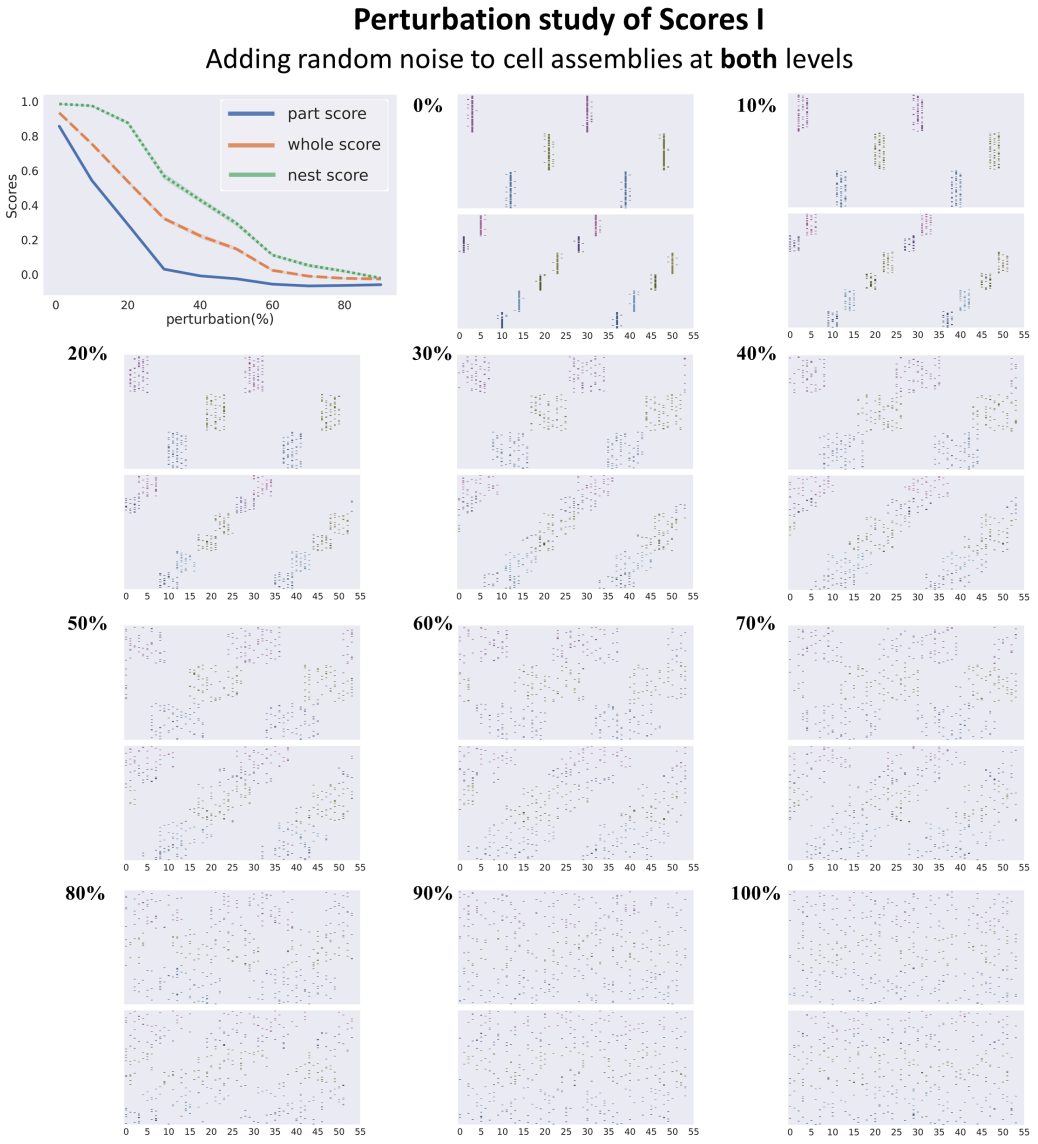

Figure 13: Perturbation study by adding increasing random noise to both part-level and whole-level spike patterns. Visualizations of the perturbed spiking pattern at different perturbation levels ($0\%to100\%$) are shown, corresponding to the Fig.5b. Both part-level and whole-level gradually degrade into random firings.

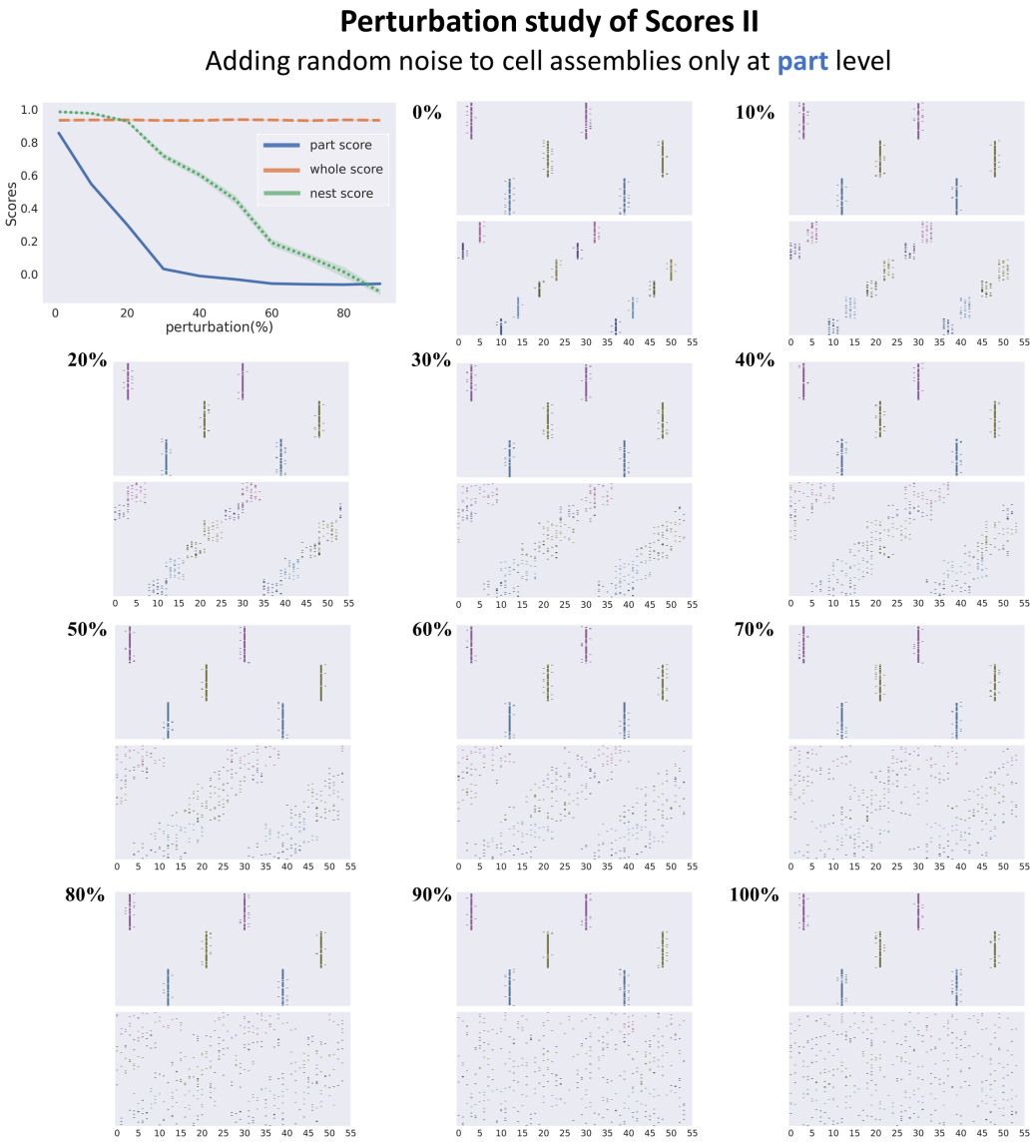

Figure 14: Perturbation study by adding increasing random noise to only part-level spike patterns. Visualizations of the perturbed spiking pattern at different perturbation levels ($0\%to100\%$) are shown, corresponding to the Fig.5c. Part-level is degraded gradually while whole level remains unchanged.

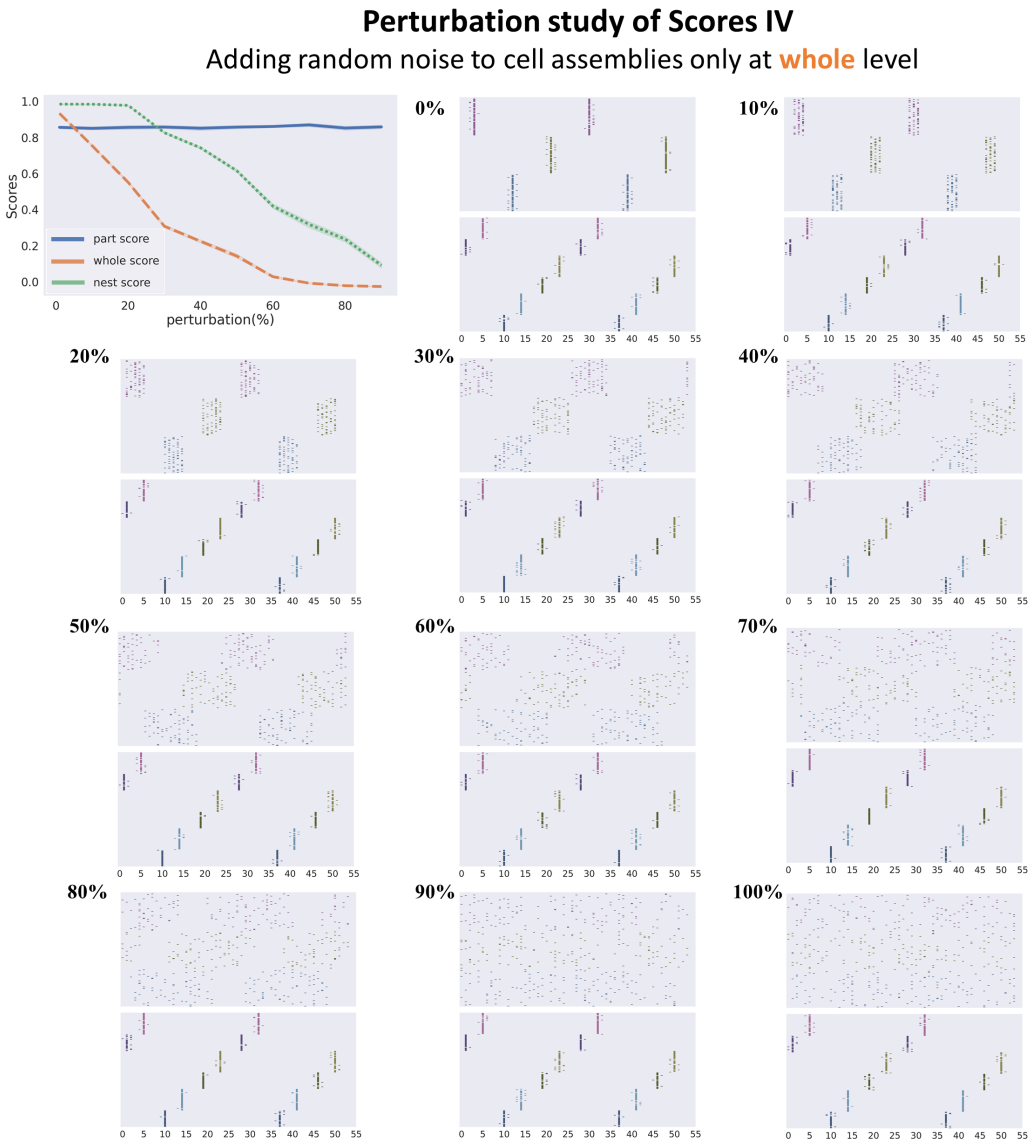

Figure 15: Perturbation study by adding increasing random noise to only whole-level spike patterns. Visualizations of the perturbed spiking pattern at different perturbation levels ($0\%to100\%$) are shown, corresponding to the Fig.5d. Whole-level is degraded gradually while part level remains unchanged.

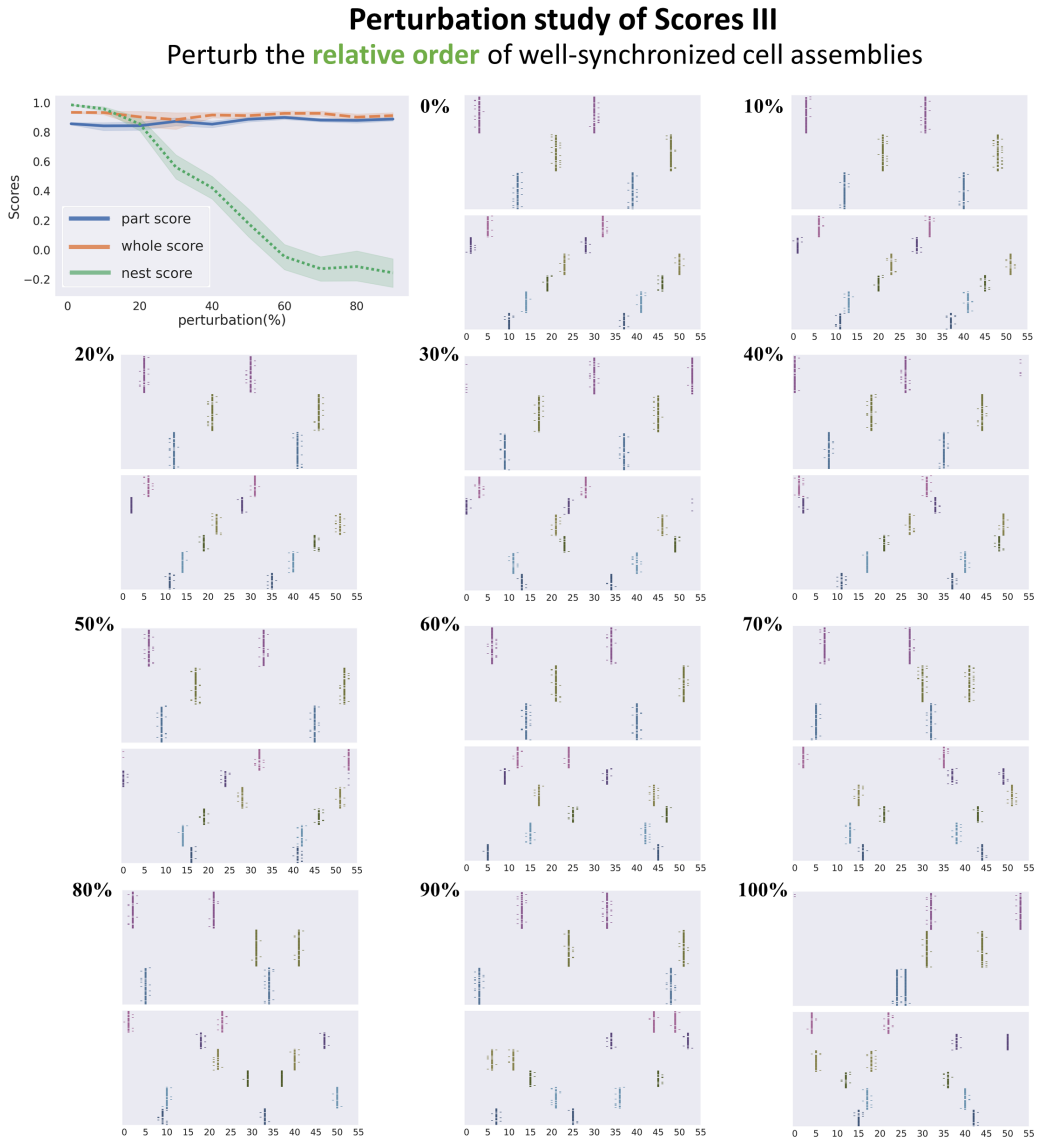

Figure 16: Perturbation study by adding increasing random noise to each assembly (both part-level and whole level). Visualizations of the perturbed spiking pattern at different perturbation levels ($0\%to100\%$) are shown, corresponding to the Fig.5e. Relative coordination among cell assemblies is gradually changed while the synchronization of each cell assembly is unchanged.

A.4 How does the Composer Work?

Mechanism overview. The nested dynamics arise from the interplay of three components, none of which suffices on its own. (1) *Recurrent DAEs* provide positive feedback: each DAE maps noisy activity toward a familiar pattern of its level, so that iterating it drives the activity toward an attractor corresponding to a single part or whole. (2) *Spiking neurons* with adaptation and transmission delays provide negative feedback: once an assembly becomes active, adaptation suppresses it, preventing the network from settling into a stationary state and allowing other assemblies to take over in turn. This winnerless competition produces sequences of synchronized assemblies, and synchrony serves as the order parameter that indicates grouping. (3) *Cross-level gating* couples the two levels: whole-level assemblies, operating on a slower timescale, gate part-level activity, so that the two levels gradually converge to a state in which each whole-level assembly is accompanied by the part-level assemblies it contains.

What is set by design and what is inferred. The number of levels, their spatial alignment, the slower timescale of the whole level, and the content priors encoded in each DAE are set by design or pretraining. Given these, the number of assemblies at each level, which features each assembly groups, and how part-level assemblies nest within whole-level ones are not specified in advance, but are inferred for each input through the dynamics above. Performance is stable across a range of timescale parameters (Fig. 18), indicating that the emergent structure is not determined by a narrowly tuned configuration.

Implementation details. The remainder of this section and the following one provide the implementation details deferred from the main text, including initialization, timescale settings, and DAE training. Additionally, the ablation studies shown in Fig. 10 are demonstrated on a single dataset for clarity; similar results on other datasets are provided here.

This section elaborates on key aspects of model implementation. Section A.4.1 and Table 4 describe the initialization of network states. Time scale parameters are detailed in Table 3 and further analyzed through ablations in Sections A.4.3 and A.4.4. Section A.5 provides DAE training procedures and associated ablations (Section A.5.5). Together, these details provide a more complete picture of how different components contribute to the emergence of hierarchical structure.

A.4.1 Initialization of the Dynamics

At the start of simulation, top-down feedback (from the DAE or higher-level SCS) is unavailable. To initialize the dynamics, we inject decaying uniform noise into the gating terms. This modifies Eqs. 1 and 4 in the main text as:

$$\rho_1(t) = x \cdot (\gamma_1 \cdot \Gamma + r_1(t) \cdot \epsilon_1)\,, \tag{21}$$

$$\rho_2(t) = (\lambda \cdot x + (1-\lambda) \cdot D) \cdot (\gamma_2 + r_2(t) \cdot \epsilon_2)\,, \tag{22}$$

where $\epsilon_i \sim \mathcal{U}[0,1]$ and $r_i(t)$ is a decaying amplitude term given by:

$$r_i(t) = r_i \cdot 0.8^{t/\tau_d}, \quad i = 1, 2. \tag{23}$$

Here, r_i is the initial noise strength and τ_d is the decay interval (Table 3). Noise is updated every τ_d steps. This mechanism enables the system to break symmetry and explore dynamics before feedback becomes structured.

A.4.2 Time scale parameters

The Composer incorporates a hierarchy of temporal integration windows, inspired by the neural syntax hypothesis [19], which suggests that the nested organization of cell assemblies reflects a nested organization of time windows. Accordingly, each spike coding space (SCS) integrates incoming spikes over a level-specific window τ_i via a leaky integrator I_i, for $i \in \{1, 2, D, \Gamma\}$. These parameters modulate the timescale of intra- and inter-column dynamics and are listed in Table 3, as used in Eq. 1–7 of the main text.

Table 3: Time scale constants of the Composer

Dataset	**Ts**	**Squares**	**SHOPs**	**Double MNIST**
T	3000	3000	3000	3000
τ_d	80	75	42	16
$\tau_{\delta 1}$	36	36	20	16
$\tau_{\delta 2}$	35	35	20	15
τ_{r1}	15	24	12	16
τ_{r2}	14	24	12	15
τ_1	2	2	3	2
τ_2	6	12	6	8
τ_D	18	30	10	8
τ_Γ	15	16	8	8
$\tau_{d'}$	80	75	42	16

Table 4: hyper-parameters of the Composer (besides time scale constants). g (in Eq.2) is the (inhibitory) gating effect of relative refractory period ($\tau_\delta - \tau_r$), same for whole-level and part-level for simplicity. λ in Eq.4 describes the skip connection. r_1, r_2 describes the initialization (Eq.23).

Dataset	**Ts**	**Squares**	**SHOPs**	**Double MNIST**
g	0.5	0.3	0.3	0
λ	0.3	0.4	0.4	0.4
noise decay	0.8	0.8	0.8	0.8
r_1	$\frac{1}{40}$	$\frac{2}{3}$	$\frac{1}{9}$	$\frac{1}{8}$
r_2	$\frac{1}{40}$	$\frac{2}{3}$	$\frac{1}{9}$	$\frac{1}{8}$

Key time constants include: τ_1, τ_2 for part and whole-level SCS integration (Eq. 3, 6); τ_D, τ_Γ for bottom-up and top-down cross-level SCS integration (Eq. 7); $\tau_d, \tau_{d'}$ for intra- and inter-level coupling delays; and $\tau_{\delta 1}, \tau_{\delta 2}, \tau_{r1}, \tau_{r2}$ for refractory periods. We set $\tau_{d'} = \tau_d$ for simplicity.

These timescales typically obey:

$$\tau_1 < \tau_2 < \tau_D \sim \tau_\Gamma \ll \tau_d, \quad \tau_{\delta 1} \sim \tau_{\delta 2} \ll \tau_d \tag{24}$$

This ordering shapes a natural temporal hierarchy, where whole-level assemblies evolve more slowly than part-level ones ($\tau_2 > \tau_1$), consistent with cortical timescale gradients [29].

If one simulation step corresponds to 1 ms, then typical biological ranges are: absolute refractory periods (τ_r) 10 ms, total refractory periods (τ_δ) 10–30 ms, coupling delays (τ_d) 50 ms [65], and coincidence detection windows (τ_i) 2–5 ms [31]. These values support gamma-band oscillatory dynamics (25–40 Hz) as commonly observed in cortical circuits [66].

A.4.3 Ablation Study of Timescale Parameters

We extend the main ablation study on timescale parameters (Fig. 10b) with further analysis across datasets (Fig. 17).

Coupling Delay. Removing feedback delay $\tau_d = \tau_{d'}$ disrupts parsing entirely by collapsing metastable attractor dynamics. Delay enables spiking neurons to recover from refractoriness before receiving feedback, stabilizing oscillatory cell assemblies [65]. This effect was formally proven for single-column dynamics in [14].

Refractory Period. Eliminating τ_{r1} or τ_{r2} prevents state transitions, causing attractors to persist and blocking oscillatory transitions (Fig. 2a→b). Without transient disruption, nested patterns do not emerge.

Integration Timescales. The constants τ_1, τ_2, τ_D shape the temporal receptive field for spike integration [19]. Small values hinder detection of distributed assemblies and impair high-level feedback.



Figure 17: Extended ablation study across datasets, expanding Fig. 10b in the main text.

Relative Refractory Gain g**.** Disabling this gain slightly reduces Nest Score under complex conditions (e.g., Ts dataset). Relative refractoriness helps avoid suboptimal attractors by enabling escape from local minima.

A.4.4 Parameter Sensitivity Test

Fig. 18 shows that coherence scores remain stable over a wide range of timescale parameters, as described in Eq. 24. These parameters serve as soft biological constraints and were not tuned. Their robustness highlights the generalizability of the Composer across settings, and supports its use as a biologically inspired framework for modeling structured vision.

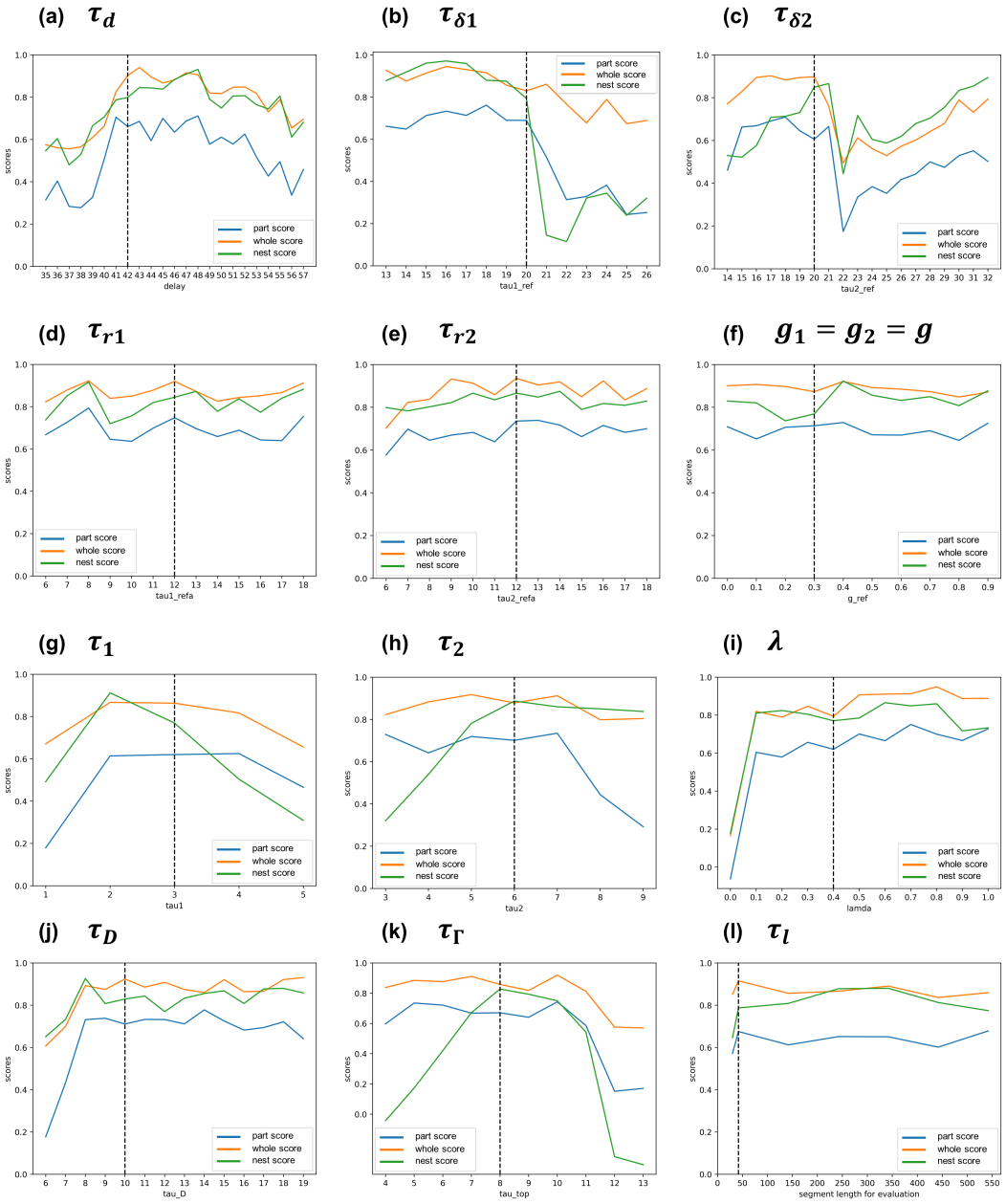

Figure 18: The sensitivity test of time scale parameters on SHOPs dataset. Black-dashed line indicates the value used for the Composer in the main text (Table 3). The values of parameters are perturbed to show how parsing degrades w.r.t parameter change. (a) The delay parameter of DAE feedback, same for part / whole level; (b) entire refractory period for part-level; (c) entire refractory period for whole level; (d)absolute refractory for part level; (e) absolute refractory for whole level; (f) the inhibitory effect of the relative refractory function; (g) the integration time window for part-level spiking neurons; (h) the integration window for whole-level spiking neurons; (i) the factor of the partial influence from skip connection; (j) the integration time window from part-level to whole level; (k) the integration time window from whole-level to part level; (l) the length of (spike train) segment used for evaluating the parsing quality (Table.2).

A.5 Implementation of Training

A.5.1 Resources

Our experiments have been performed on ubuntu 16.04.12 with devices: CPU (Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.4GHz) and 4×GeForce RTX 2080 Ti. The python version is 3.6.3.

A.5.2 Network architecture and training hyperparameters

The details of training neural networks are shown in Table.5. All networks are trained with stochastic gradient descent (SGD).

Table 5: Details of training DAE

Dataset	**encoder**	**decoder**	**learning rate**	**noise**	**minibatch size**	**epoch num**
Ts (part)	FC(1600, 1000) Sigmoid()	FC(1000, 1600) Sigmoid()	1e-3	0.5	16	200
Ts (whole)	FC(1600, 1000) Sigmoid()	FC(1000, 1600) Sigmoid()	1e-3	0.5	16	200
Squares (part)	FC(3600, 400) Sigmoid()	FC(400, 3600) Sigmoid()	1e-3	0.8	16	200
Squares (whole)	FC(3600, 400) Sigmoid()	FC(400, 3600) Sigmoid()	1e-3	0.6	16	200
SHOPs (part)	FC(3600, 400) Sigmoid()	FC(400, 3600) Sigmoid()	1e-3	0.7	16	200
SHOPs (whole)	FC(3600, 400) Sigmoid()	FC(400, 3600) Sigmoid()	1e-3	0.7	16	200
Double-MNIST (part)	FC(6400, 2000) Sigmoid()	FC(2000, 6400) Sigmoid()	1e-3	0.5	16	200
Double-MNIST (whole)	FC(6400, 2000) Sigmoid()	FC(2000, 6400) Sigmoid()	1e-3	0.5	16	200

A.5.3 Dataset for Training the DAE

Details of the training dataset are provided in Table 6, with visual examples shown in Fig. 19. The dataset design follows the convention established in [14].

Table 6: Training dataset details

Dataset	**Training size**	**Input dimension**	**# Object**
Ts (part)	60000	40×40	1
Ts (whole)	60000	40×40	1
Squares (part)	60000	60×60	1
Squares (whole)	60000	60×60	1
SHOPs (part)	20000	60×60	1
SHOPs (whole)	20000	60×60	1
Double-MNIST (part)	60000	80×80	1
Double-MNIST (whole)	60000	80×80	1

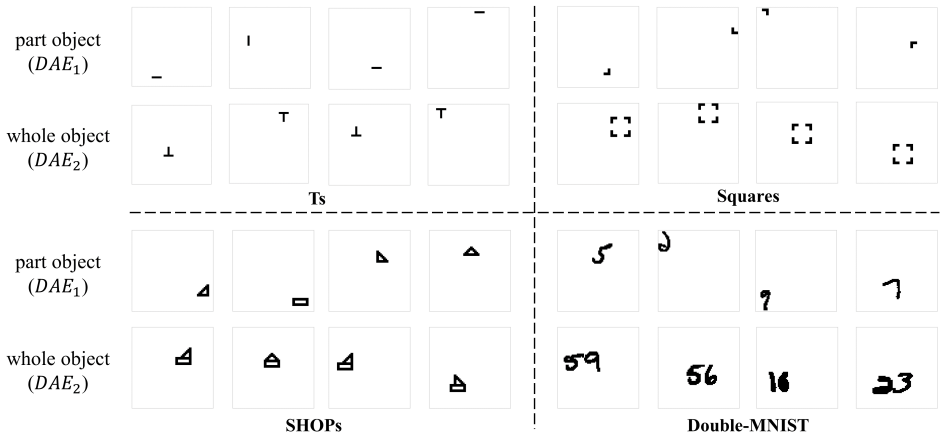

Figure 19: Examples of the training data to train part/whole level DAE.

A.5.4 Loss Function

The DAE modules (part and whole) are trained using a standard Binary Cross Entropy (BCE) loss between the denoised output and the original binary object image:

$$\text{loss}(x) = \text{BCE}(x, \text{DAE}_i(\tilde{x})), \quad i = 1, 2 \tag{25}$$

where x is a single-object image (Fig. 19) and $\tilde{x}$ is its corrupted version. This unsupervised loss encourages the DAE to learn content-level priors for part and whole objects, without providing any explicit structural information. The part-whole hierarchy and grouping structure emerge entirely from the model's dynamics during simulation.

Such prior knowledge reflects an inductive bias about what individual objects (e.g., eyes, nose, mouth) look like—an assumption that is both biologically plausible and essential for interpreting multi-object scenes. While some priors may be evolutionarily encoded, others may be acquired during learning and development.

Recent work [61] suggests a path toward end-to-end unsupervised learning. The model's architectural constraints—hierarchical column structure, delay-coupled feedback, and timescale organization—act as inductive biases for structural learning. These mechanisms bias the system toward forming nested assemblies and selectively predicting objects in accordance with their temporal resolution.

Rather than pretraining DAE modules, one could treat the entire system as a recurrent spiking neural network and optimize it via backpropagation through time (BPTT) [67]. A minimal modification would replace the DAE pretraining with a reconstruction loss between the input scene and the averaged top-down signals (e.g., $\gamma_1(t)$ or $\Gamma(t)$).

Inspired by prior work [61], this approach may allow the model to learn both object content and compositional structure in a fully unsupervised setting—supporting the hypothesis that neural syntax can emerge through predictive interaction with the environment [62].

A.5.5 Ablation study of the DAE

We assess the relationship between DAE denoising quality and parsing performance by evaluating 100 randomly initialized DAE pairs per dataset. For each pair, one part-level and one whole-level DAE are trained using a randomly sampled learning rate from $(10^{-3}, 1)$. The x-axis in Fig. 10a shows the combined training loss of both DAEs, and the corresponding Composer scores are plotted on the y-axis. Outlier models with divergent losses due to unstable learning rates are excluded.

As shown in Fig. 10a, and consistently across datasets (see also Fig. 20), lower denoising loss correlates with higher parsing scores, regardless of which DAE loss is used (Fig. 20 (a) vs (b,c)). This supports the finding presented in Fig. 10a as representative.

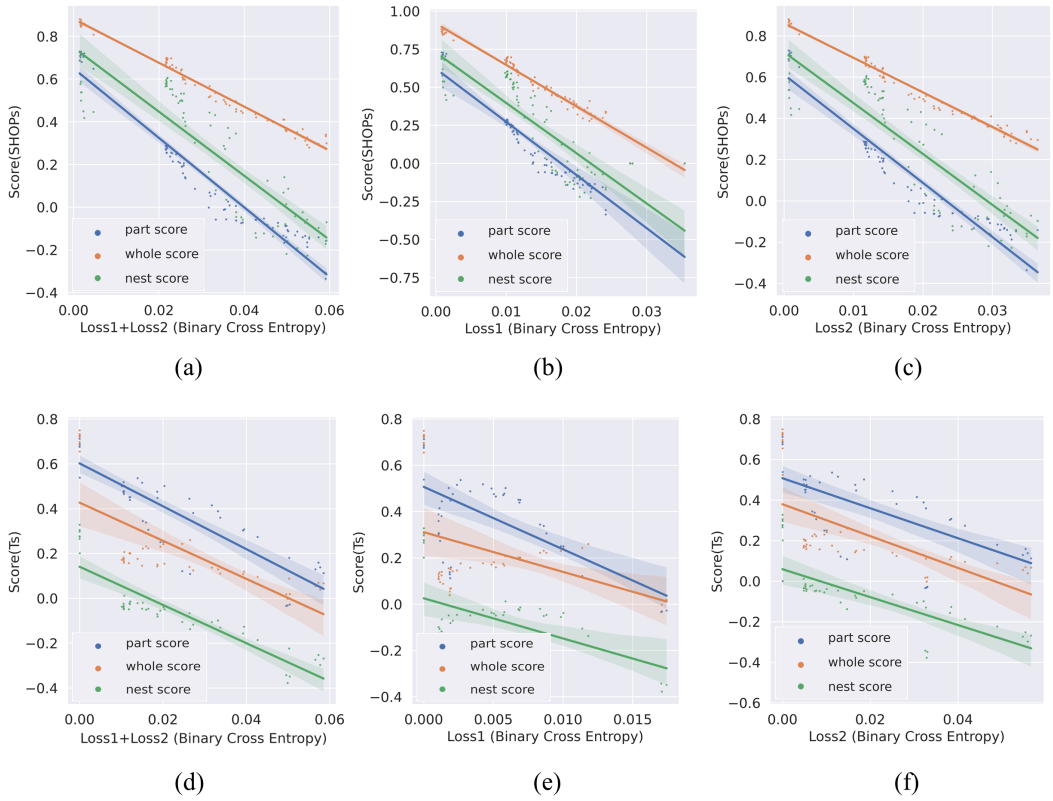

Figure 20: Loss vs Score (additional results). (a)(b)(c) results on SHOPs dataset; (d)(e)(f) results on Ts dataset. (a)(d) Relations between scores and total loss of part / whole level DAE ($loss_1 + loss_2$); (b)(e) Relations between scores and total loss of part-level DAE ($loss_1$); (c)(f) Relations between scores and loss of whole-level DAE ($loss_2$); All results are consistent.

These results confirm that DAE quality directly impacts the Composer's structure emergence. Better denoising improves the fidelity of associative memory (Fig. 2a), which in turn supports the formation of coherent, hierarchical assemblies (Fig. 2c).

A.6 Relation to Neuroscience

This section outlines the neuroscience principles that motivate the design of the Composer. For each principle, we first describe the biological concept and then explain how it is reflected in the model. These correspondences are drawn at an abstract level and serve as inspiration rather than as validated claims: several underlying ideas, such as cell assemblies, binding by synchrony, and neural syntax, remain hypotheses under active study. We therefore also discuss competing evidence and alternative views at the end of this section.

A.6.1 Circuit architecture.

- **Feed-forward and feedback pathways along the cortical hierarchy.** The visual cortex is organized as a hierarchy of areas linked by reciprocal connections. Within each area, neurons are often described as organized into columns spanning six layers (I to VI; Fig. 2d,g) [68], although the functional significance of the column remains debated [69]. Connections between areas follow characteristic laminar patterns: feedforward projections, which carry signals up the hierarchy, and feedback projections, which carry signals down, originate and terminate in different layers. Markov et al. [70] anatomically mapped such feedforward and feedback connections between layers II/III of lower- and higher-level areas, and further identified long-range feedforward connections through layer IIIb. Cortical feedback more generally includes signals from both superficial layers (II/III) and deep layers (V/VI). In the Composer, the interaction between part-level and whole-level SCS

layers parallels this laminar-specific communication between areas, and the skip connections from the external input x to the whole-level SCS (Eq. 4) parallel the long-range feedforward pathway. The SCS and the DAE latent spaces loosely correspond to superficial and deep layers, respectively; investigating cross-level interactions between the SCS and the latent spaces, beyond the current pixel-level gating, is left for future work.

- **Topographic alignment.** A defining feature of sensory cortex is its topographic organization: neighbouring neurons respond to neighbouring locations in sensory space, forming maps of the sensory surface, such as retinotopic maps in visual cortex [63, 71]. In the visual system, these maps are preserved across successive areas of the hierarchy (e.g., V1, V2, V4), with receptive fields growing progressively larger, so that a given location in a higher area covers a broader region of the same visual field. Topography thus provides a shared spatial reference frame through which representations at different levels refer to the same locations in the scene, and how such maps form has recently been modeled with end-to-end topographic networks [72]. The Composer adopts this principle by aligning neurons across levels with the input image: a part-level and a whole-level neuron at the same location refer to the same point in the scene, while the whole level groups features over larger regions. This shared reference frame makes hierarchical inclusion directly interpretable, analogous to the "identical islands of vectors" in GLOM [4].
- **Dendritic modulation and gating.** Cortical pyramidal neurons do not simply sum their inputs: their dendrites perform nonlinear integration [73]. Bottom-up sensory input arrives mainly at basal and proximal dendrites, whereas top-down feedback from higher areas targets the distal apical dendrites in superficial layers. Coincident input to the two compartments can trigger dendritic spikes and burst firing, so that feedback multiplicatively amplifies the response to feedforward drive rather than adding to it [74]. This coupling has been proposed as a general mechanism by which the cortex associates internal predictions with external input [75]. The Composer abstracts this principle as a multiplicative gating mechanism within each column and across levels, in which top-down signals modulate bottom-up drive; this gating is essential for structured assembly formation.
- **Recurrent connectivity.** Local recurrent connections are a prominent feature of cortical circuits: a large fraction of the synapses a cortical neuron receives come from nearby neurons within the same area, and these recurrent interactions shape, normalize and amplify responses to external input [68, 76]. The SCS in the Composer omits explicit intra-layer recurrence as a simplification. Instead, the DAE can be viewed as parameterizing effective recurrence via its latent projections, analogous to how recurrent interactions can be implemented by additional modulatory elements in neuron-astrocyte networks [77].

A.6.2 Dynamics.

- **Delay-coupled recurrent dynamics.** Signals between cortical neurons are not transmitted instantaneously: axonal conduction and synaptic transmission introduce delays ranging from a few to tens of milliseconds. The cortex can therefore be described as a network of delay-coupled oscillators [65], whose activity depends not only on its current state but also on its recent history. Such delays allow the network to generate a large repertoire of reproducible spatiotemporal firing patterns [64]. The Composer incorporates delay-coupled recurrence, which plays a critical role in its dynamics (Section A.4.3). The coupling delay introduces a temporal window that allows transient cell assemblies to compete and emerge sequentially, rendering the system non-Markovian and substantially expanding its effective state space. As a result, stable representations are realized as dynamic trajectories rather than fixed points, potentially expanding the system's capacity for associative coding.[2]
- **Metastability of cortical networks.** Single-trial analyses of ensemble activity have shown that cortical dynamics do not settle into a single stable state, but evolve through sequences of metastable states. Such metastable dynamics have been implicated in sensory coding, memory, and decision-making, although the network mechanisms that generate them remain unclear [78, 79]. The Composer offers one mechanism by which metastable dynamics can emerge, through the interaction between spiking neurons and denoising autoencoders, and illustrates their computational role in vision (Fig. 2).

[2]Each memory corresponds to a trajectory composed of transiently stable states, allowing a combinatorially larger set of attractor-like representations.

- **Timescale gradient.** Cortical areas differ systematically in their temporal dynamics: along the hierarchy from early sensory to higher-order areas, dominant oscillation frequencies decrease [29] and intrinsic timescales of neural activity lengthen [80]. This gradient has been explained by differences in local recurrent connectivity across areas [81], and is thought to allow higher areas to integrate information over longer time windows. The Composer leverages this temporal hierarchy by assigning a slower timescale to the whole level than to the part level, which forms the basis for representing part-whole inclusion through nested cell assemblies. Since the available range of timescales is finite, it also constrains the number of reliably separable levels. We view this as a limitation shared by biological and computational systems, consistent with the bounded capacity of human working memory [82, 83] and with the view that a small number of levels suffices for part-whole parsing [4]. Future extensions may explore explicit frequency-based separation using gamma, alpha, and theta bands to support deeper or more flexible hierarchies.
- **Gamma oscillation and perceptual coherence.** Gamma-band oscillations (roughly 30–80 Hz) arise from interactions between excitatory and inhibitory neurons and are prominent in visual cortex during stimulus processing. Rhythmic gamma-band synchronization has been proposed to link distributed neuronal activity into coherent percepts [66]. If one timestep is taken as 1 ms, the spiking dynamics of the SCS approximate gamma-frequency rhythms, and these oscillations dynamically assemble neurons into object representations, consistent with the representation hypothesis in the main text. The functional role of gamma synchrony in binding, however, remains debated (see Open debates below).
- **Coincidence detection.** Cortical neurons can act as integrators, summing inputs over long time windows, or as coincidence detectors, responding preferentially to inputs that arrive nearly simultaneously. With short integration windows, neurons become sensitive to the precise timing of their inputs, so that synchronous spikes are transmitted more effectively than the same number of asynchronous ones [84]. The SCS uses short integration windows, enabling the temporal resolution of synchronous events.
- **Polychronization.** Because conduction delays vary across connections, spikes emitted by different neurons at different times can arrive at a common target simultaneously. Groups of neurons can therefore fire in precise, reproducible patterns that are time-locked but not synchronous, a phenomenon termed polychronization [64]. In the Composer, coupling delays may cause part and whole assemblies to appear offset in time. From the perspective of a readout neuron, however, their arrival times are synchronized, so temporal offsets preserve coherence under a polychronous interpretation [19].

A.6.3 Representation.

- **Cell assemblies and neural syntax.** The cell assembly hypothesis proposes that cognitive entities are represented by transiently co-active groups of neurons rather than by single cells [20]. Building on this idea, Buzsáki proposed that assemblies are organized into sequences nested within oscillations of different frequencies, forming a "neural syntax" analogous to how letters form words and words form sentences [19]. Neural sequences have been observed in many brain regions, including visual cortex [85–87], but how they are organized and what they encode remain open questions, and whether they represent part-whole structure has not been directly tested. The Composer operationalizes this hypothesis by forming nested temporal structures of assemblies via synchrony and sequencing, in which assemblies may serve as transient code words for compositional representation.
- **Temporo-spatial theory of consciousness.** Northoff [88] proposed that temporo-spatial nestedness of neural activity underlies the state or level of consciousness, while temporo-spatial alignment between neural activity and the environment shapes its content and form. In our framework, nestedness may reflect perceptual awareness (e.g., recognizing part-whole structure), and alignment may capture scene structure. As a speculative extension, we hypothesize that conscious awareness emerges when part-whole organization is well formed in internal representations (neural syntax), and that the quality of these structures, reflected in neuronal coherence, determines the clarity of perception, an idea that remains to be tested.
- **Sparse and dense coding.** Cortical circuits have been reported to employ dual coding schemes: ultra-sparse firing in superficial layers (II/III), where only a small fraction of neurons respond to a given stimulus, and denser activity in deeper layers (V/VI) [89, 90]. Dense codes may capture statistical regularities of features, whereas sparse codes may support relational or contextual

encoding. In the Composer, this duality is reflected by the sparse spike-based SCS and the dense real-valued latent codes of the DAE, with synchrony within the SCS supporting relational structure among objects.

A.6.4 Learning and priors.

- **Predictive coding and autoencoding.** Predictive coding proposes that higher cortical areas send predictions of lower-level activity through feedback connections, while lower areas send forward only the prediction errors that remain unexplained. The framework was originally proposed by Rao and Ballard to explain extra-classical receptive field effects [91], and was later generalized into a cortical canonical microcircuit model [92] and a unified theory of brain function [93]. At an abstract level, bottom-up and top-down signaling in this framework can be viewed as an autoencoder, in which minimizing reconstruction error corresponds to minimizing prediction error or surprise. The Composer adopts this abstraction, implementing within-column bottom-up and top-down pathways with a denoising autoencoder (DAE). The resulting recurrent loop also gives rise to temporal coherence in the Composer, echoing coherent activity observed in cortex [37]. More broadly, the DAE illustrates a modeling strategy in which a component of the cortical circuit is represented by a trainable network whose function is learned from data rather than specified by hand. Such data-driven components complement classical, theory-driven circuit models and may suggest hypotheses that these models would not reach [94, 95].
- **Preconfigured brain.** In Buzsáki's "inside-out" framework, the brain is a preconfigured system: a genetically specified network equipped with a repertoire of initially meaningless activity patterns and internally generated syntactical rules, which acquire meaning through experience [32]. In this view, perception does not build representations from scratch; rather, external input is matched to preexisting neural patterns, and hierarchically nested rhythms (neural syntax) shape the structure of the resulting neural messages. The pretrained DAEs in the Composer play the role of such preconfigured wiring: they store level-specific prototypes of parts and wholes (Sec. A.1.5), acquired through evolution or life-long sensorimotor calibration. Given a new stimulus, how the scene is segmented and its structure inferred depends on how the input is matched to these prototypes through the recurrent dynamics, with each emerging assembly corresponding to a portion of the input matched to a prototype. In this view, plasticity would primarily serve to refine this "good-enough" model through experience; incorporating such ongoing plasticity into the Composer is left for future work.
- **Plasticity.** Synaptic learning in the brain is commonly described by Hebbian mechanisms, in which connections between co-active neurons are strengthened, and by their spike-timing-dependent variants [76]. The DAE in the Composer is instead trained via gradient descent, which can be reconciled with this view in two ways. First, if the DAE is interpreted as an evolved structure, gradient-based optimization can be viewed as a stand-in for evolutionary search. Second, the reconstruction loss in DAE training promotes alignment between input and modulatory feedback, echoing correlation-based plasticity; in simple autoencoders, gradient descent and Hebbian learning have been formally connected [96]. More generally, the Composer separates two processes that may correspond to different brain states. During wakefulness, the DAE weights are fixed and the network infers the structure of the current input through its dynamics; during offline periods such as sleep, replayed activity could be used to update the DAE weights, playing the role that DAE training plays in our model. Such a separation between fast inference and slower offline learning is also discussed in GLOM [4].

A.6.5 Open debates and alternative views.

- **Binding by synchrony.** The temporal binding hypothesis proposes that neurons encoding features of the same object fire synchronously, so that spike timing marks which features belong together [23, 97, 98]. The hypothesis has been highly influential, but has also been criticized on theoretical grounds [24], and several studies found that synchrony does not track perceptual grouping, for example in motion coherence in area MT [99] or in contour grouping in V1, where grouping was instead reflected in enhanced firing rates [100].
- **Alternative binding mechanisms.** Incremental grouping proposes that features are bound by spreading enhanced firing rates across neurons that represent the same object, a process closely linked to attention [26, 101]; feature integration theory likewise emphasizes the role of attention in binding [25]. Notably, the Composer combines both ingredients: top-down attention signals

Table 7: Scores of the Composer

Dataset	**Ts**	**Squares**	**SHOPs**	**Double MNIST**
Part Score	0.73 ± 0.005	0.68	0.67 ± 0.008	0.17 ± 0.003
Whole Score	0.69 ± 0.001	0.95	0.87 ± 0.003	0.39 ± 0.005
Nest Score	0.28 ± 0.004	0.69	0.82 ± 0.004	0.39 ± 0.002

from the DAEs ($\gamma_1, \gamma_2, \Gamma$) co-emerge with spike synchrony and jointly support grouping. This co-emergence of attention and coherence suggests that rate-based labeling and temporal coherence may be complementary rather than mutually exclusive.

- **Slot-like codes and modular organization.** Slot-based models assign each structural unit to one of a fixed set of discrete slots. It is unclear, however, how a continuous, distributed neural substrate would specify such slots and the boundaries between them, and coherence-based grouping does not require them. Although this remains largely an open question, recent studies suggest that slot-like representations may not be entirely implausible in the brain, and we include them here for completeness. In mouse frontal cortex, neurons tuned to progress through structured behavioural sequences form organized activity patterns that resemble structured memory buffers, in which different positions of a sequence are held in distinct neural populations [102]. Moreover, discrete modules with sharp boundaries can emerge from local interactions and smooth gradients [60], suggesting that such discreteness may itself arise from continuous mechanisms. Even so, slot-based models typically require all slots to share the same weights, so that any object can be represented in any slot [6], and this requirement is not easy to realize biologically, especially when learning through experience is considered: an update to the weights of one slot would have to be copied exactly to all other slots, and it is hard to envision a biological mechanism that could achieve this.

  Our own model is not entirely free of slots: its discrete levels can be viewed as slots, assigned to levels of the hierarchy rather than to individual objects. This is a much weaker requirement, because levels are more abstract than objects and far fewer of them are needed. A scene may contain many objects, but the levels that organize them are typically few. Following the insight on modular emergence [60], these discrete levels might also be made continuous in the future, for example by letting them emerge from gradients in parameters such as timescales (Sec. A.1.2.1).

These debates concern how the brain actually binds features, a question that remains unresolved, and our work does not take a side on it. Instead, we use neuronal coherence as a computational principle for representing discrete part-whole structure in a continuous, distributed form in neural networks. Whether the brain relies on the same principle, and how the competing views above should be weighed, is beyond the scope of this work.

A.7 Benchmarking

A.7.1 Composer scores (corresponding to Fig.9)

Table 7 reports the mean scores of the Composer across 1000 randomly selected samples, averaged over 5 random seeds. These values correspond to the benchmark results shown in Fig. 9 of the main text. Error bars are negligible, indicating stable average performance across random seeds (for Squares, the raw results were lost; means are read from Fig. 9, and standard deviations are not reported but are below 0.01 according to the error bars in Fig. 9).

A.7.2 Agglomerator scores (corresponding to Fig.9)

Table 8 reports the mean scores of the Agglomerator across 1000 randomly selected samples, averaged over 5 random seeds. These results correspond to the benchmark comparison in Fig. 9 of the main text. Error bars are minimal, indicating consistent average performance across random seeds.

A.7.3 Introducing the Agglomerator

The Agglomerator [17] is a GLOM-inspired model [4] designed to represent part-whole hierarchies through spatially organized neuronal coherence. It groups vector-based representations, referred to as "columns," into clusters of similar vectors ("identical islands of vectors"), where spatial nesting

Table 8: Scores of the Agglomerator

Dataset	**Ts**	**Squares**	**SHOPs**	**Double MNIST**
Part Score	-0.24 ± 0.002	-0.35 ± 0.002	-0.25 ± 0.001	-0.18 ± 0.003
Whole Score	-0.28 ± 0.002	-0.16 ± 0.001	-0.18 ± 0.002	0.00 ± 0.001
Nest Score	-0.40 ± 0.003	-0.37 ± 0.003	-0.36 ± 0.001	0.01 ± 0.004

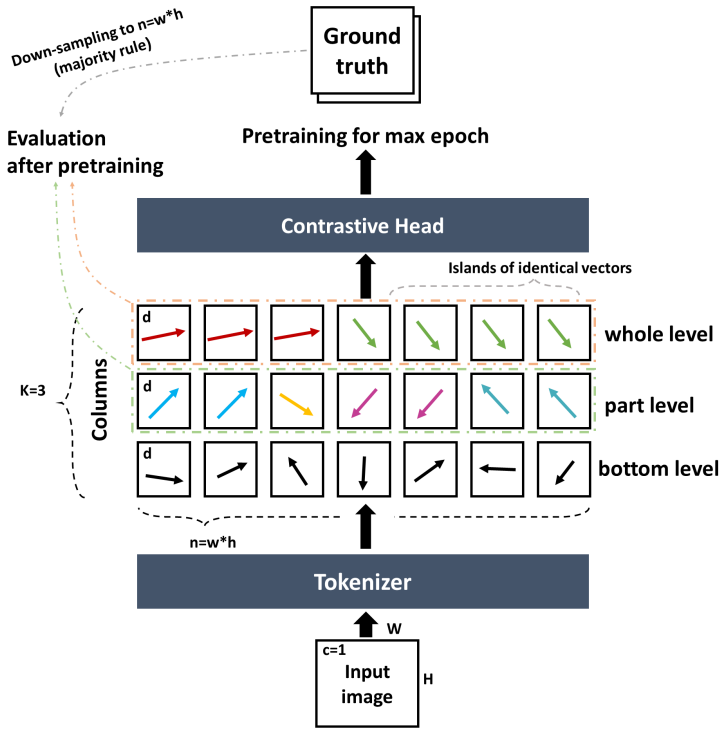

Figure 21: Illustration of the architecture of the Agglomerator: How part-whole hierarchy should be represented as spatially nested islands of identical vectors and how evaluation is achieved.

across levels is interpreted as a hierarchical structure. Figure 21 illustrates its architecture and representational scheme for reference. Columns are arranged on a spatial grid across multiple levels, with each level operating at a distinct spatial scale determined by the radius of horizontal connections. The Agglomerator was selected as a benchmark due to its conceptual similarity with Composer—it also aims to form level-wise structured representations via coherence and spatial inclusion. However, as shown in Table 8, it fails to produce coherent node-level groupings in all four datasets, as reflected by near-zero Part and Whole Scores. This suggests a lack of effective intra-level assembly formation. Furthermore, because GLOM-style architectures encode hierarchical coordination implicitly through spatial containment rather than cross-level representational similarity, the Nest Score offers limited interpretability and remains lower than that of Composer.

A.7.4 How is coherence measured in the Agglomerator?

Actually, it is direct to generalize the coherence metrics to account for evaluating the identical islands:

$$\text{Part Score} = \text{Silhouette}(D_2(l_p, l_p), \text{label}_1) \tag{26}$$

$$\text{Whole Score} = \text{Silhouette}(D_2(l_w, l_w), \text{label}_2) \tag{27}$$

$$\text{Nest Score} = Silhouette(D_2(l_p, l_w), \text{label}_2) \tag{28}$$

where D_2 is the Euclidean distance metric, $l_p \in \mathbb{R}^{(w*h,d)}$ is the part-level column representation. w and h is the size of the grid mesh of columns, d is the dimension of each column. Here, the dimension d is parallel to the time dimension in the Composer. If identical islands are formed, Part / Whole Scores should reach values near 1.

A.7.5 The architecture detail of the Agglomerator

We closely follow the architectural and hyperparameter settings provided in the original paper. Detailed parameters are listed in Table 9, with schematic explanations shown in Fig. 21. We set the number of levels $K = 3$, where the bottom level (level 1) corresponds to the output of the tokenizer and encodes extracted features only. Thus, we interpret level 2 as the part-level and level 3 as the whole-level. The number of inference iterations is denoted by T (Table 9).

Table 9: Parameter details for benchmarking

Parameter	**Ts**	**SHOPs**	**Squares**	**Double-Digit MNIST**
K	3	3	3	3
d	48	48	48	125
T	5	5	5	5
w,h	10	15	15	8
c	1	1	1	1
max epoch	100	100	100	300
batch size	32	32	32	128
learning rate	0.05	0.05	0.05	0.05
W, H	40	60	60	80
number of training objects	6	3	3	2

Let d denote the dimensionality of each column's representation, and w, h the spatial dimensions of the column grid, shared across all levels. The total number of columns is $n = w \times h$. As the datasets are binary, the input channel c is set to 1. W and H represent the input image dimensions. The model is trained for a maximum of 100 epochs, consistent with the pretraining setup in the original implementation. All evaluations are performed after 100 epochs of contrastive pretraining.

Table 10 describes the architecture of the convolutional tokenizer. The embedding dimension e_d is set to 12, 12, 5, and 12 for SHOPs, Squares, Double-Digit MNIST, and Ts datasets, respectively.

The tokenizer output is reshaped and fed into the bottom-up and top-down column networks, whose structures are provided in Table 11. For the bottom-up network, d_1 is set to 96, 96, 250, and 96, while d_2 is 384, 384, 1000, and 384, across the four datasets. For the top-down network, d_1 is set to 48, 48, 125, and 48, and d_2 to 192, 192, 500, and 192. The number of patches n_p matches $n = w \times h$ and takes values 225, 225, 64, and 100 for SHOPs, Squares, Double-Digit MNIST, and Ts, respectively.

Finally, the structure of the contrastive learning head is shown in Table 12. The patch dimension p_d is set to 48, 48, 125, and 48, aligned with the corresponding datasets. The number of patches n_p remains consistent with that used in the column networks.

A.7.6 Down-scaling of the ground truth

In the Agglomerator, each input image of size $W \times H$ is tokenized into $n = w \times h$ patches, where $w = W/4$ and $h = H/4$. These patch embeddings form the representations of the bottom-level columns. Due to this down-sampling, the number of columns is significantly smaller than the number of input pixels. Moreover, the computational complexity of the Agglomerator scales as $O(w^4)$,

Table 10: Parameter details for convolution network

Network Structure
Conv2d(1, e_d // 2, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1), bias=False)
BatchNorm2d(e_d // 2)
ReLU()
Conv2d(e_d // 2, e_d // 2, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
BatchNorm2d(e_d // 2)
ReLU()
Conv2d(e_d // 2, e_d, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
BatchNorm2d(e_d // 2)
ReLU()
MaxPool2d(kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)

Table 11: Parameter details for column network

Network Structure
LayerNorm(n_p)
Conv1d(d_1, d_2)
GELU (for bottom-up network) or Siren (for top-down network)
LayerNorm(n_p)
Conv1d(d_2, d_1)

making it infeasible to preserve the original spatial resolution (i.e., $w = W$, $h = H$). To maintain consistency between model outputs and evaluation labels, we apply the same down-sampling ratio to the ground truth annotations. The resulting $w \times h$ label map aligns with the column grid and enables per-column evaluation. Down-sampling is performed using a majority-vote rule within each 4×4 region.

A.7.7 Limitations of the Agglomerator under structured evaluation

As shown in Table 8, the Agglomerator fails to produce meaningful hierarchical representations across all benchmark datasets. This outcome is not entirely unexpected: in its original formulation [17], the model was evaluated solely on classification accuracy, without direct assessment of hierarchical structure. As a result, it remains unclear whether the model actually instantiated the structural expectations outlined in GLOM [4]. Visualizations further suggest that its multi-level representations primarily reflect feature extraction at different spatial scales, rather than interpretable part-whole decomposition. While the underlying representation hypothesis—grouping by neuronal coherence—is conceptually compelling, it has not been quantitatively validated or falsified under structure-sensitive conditions.

While the most likely reason for the failure is that the Agglomerator lacks the capacity to form explicit hierarchical representations, other contributing factors may also be relevant. One such factor is dataset symmetry. In the original Agglomerator experiments, the input images contained color differences between parts, and visualizations suggest that grouping was often based on local appearance similarity—for example, identical patches with similar colors. If this low-level feature matching serves as a primary grouping mechanism, it is unlikely to generalize to our binary datasets, where all objects share the same color (black). In this setting, grouping must instead arise from internal symmetry-breaking mechanisms within the model. This highlights a deeper challenge: Because all foreground pixels in binary scenes share the same value, low-level similarity cues such as color or texture cannot separate objects. Parsing part-whole structure in such scenes therefore relies on organizational cues, such as spatial proximity and connectedness, and on part-whole consistency across levels.

We emphasize that the failure of the Agglomerator in our benchmarks does not imply that the model—or its underlying design principles (GLOM [4])—is fundamentally flawed. The GLOM-inspired architecture remains a flexible and promising framework for hierarchical representation, particularly in real-world contexts. However, our evaluation suggests that the current implementation

Table 12: Parameter details for contrastive head

Network Structure
LayerNorm(p_d)
Dropout(p=0.3)
Rearrange('b n d - b (n d)')
LayerNorm(n_p * p_d)
Dropout(p=0.3)
Linear(n_p * p_d, n_p * p_d)
LayerNorm(n_p * p_d),
GELU()
LayerNorm(n_p * p_d)
Dropout(p=0.3)
Linear(n_p * p_d, 512)

of the Agglomerator is not yet sufficient to learn part-whole structure from data. Achieving this may require further refinement of the loss function, architectural constraints, or other components critical to inducing hierarchical organization.

These results reinforce the broader point: evaluating part-whole hierarchy requires dedicated metrics and datasets. Models should be assessed not only by task success, but also by their internal structural coherence and interpretability.

A.8 Additional experimental details and results

A.8.1 Convergence behavior across datasets

Figure 8 shows the convergence of Part, Whole, and Nest Scores over time. Each curve is computed by averaging over 100 randomly selected samples, with scores evaluated every 100 time steps (i.e., 30 evaluations over the full 3000-step simulation).

SHOPs. Convergence on the SHOPs dataset is the most stable overall. However, overlapping part-level objects within a whole introduce ambiguity, resulting in a relatively lower Part Score (Fig. 8a).

Squares. This dataset reveals an interesting temporal pattern. The Whole Score rises rapidly and consistently leads the other scores, suggesting that global structure is recognized before local parts—consistent with findings from human vision and Gestalt principles [34]. In contrast, the Part and Nest Scores initially decline before recovering. This can be explained by a phase transition: once square-like wholes begin to emerge (around time step 500–1000), they impose new global constraints that temporarily disrupt part-level assemblies—e.g., reducing firing density or introducing transient misgroupings. After this brief adaptation period, both scores recover and converge.

Ts. Parsing is particularly challenging on this dataset due to the larger number of objects (6 wholes, 12 parts). This leads to a combinatorial explosion in potential part-whole groupings (e.g., 6^{12} configurations). Although the model can establish local coherence, aligning parts and wholes into valid nested structures requires significant search and coordination. As a result, the Nest Score lags behind. Since neural computation in the brain has been proposed to operate as a sampling process [103], the combinatorial complexity involved in coordinating part-whole relationships may also contribute to delayed or unstable perception—analogous to challenges observed in phenomena such as the binding problem [98, 104].

Double-Digit MNIST. Diversity in object shapes and configurations presents another challenge. The model struggles to consistently distinguish and group digits, resulting in lower Part Scores and higher variance. Nonetheless, the Composer eventually converges to a coherent part-whole structure, as evidenced by the stable Nest Score—suggesting robust hierarchical inference despite high input variability.

A.8.2 Varying the Number of Objects Across Datasets

Setup. To test whether the emergent structure adapts to scenes of different complexity, we vary the number of objects per scene on each of the four datasets, choosing a range appropriate to each dataset (Table 13). As in the main experiments, the DAEs are trained only on single part or whole objects (Table 6), while all other parameters of the Composer are kept the same as in the main experiments (Tables 3 and 4). The same model is then applied, without any retraining or parameter tuning, to scenes with different numbers of objects. Scenes are generated with the same procedure as the original datasets, except for the number of objects placed on the canvas. For each object count, we evaluate 1000 scenes and repeat the simulation with 5 random seeds, reporting the mean and standard deviation of the Part, Whole, and Nest Scores across seeds.

Results. Table 13 summarizes the results. On SHOPs, Squares, and Ts, the number of emergent assemblies and their nesting change with the number of objects, and the scores decrease only gradually as scenes become more crowded; the largest drops occur when heavy overlap on the fixed canvas makes the scenes themselves ambiguous.

Double-Digit MNIST behaves differently. Because handwritten digits vary considerably in shape and stroke width, this dataset is the most challenging for the current model even in its original setting, as reflected by its lower scores with two objects. Adding more objects further increases clutter and overlap, and given the limited capacity of the current minimal DAE (Sec. A.1.2.2), the model largely fails to recover the part-whole structure with four objects, where the scores approach zero. We expect stronger DAE architectures to alleviate this limitation.

A.8.3 Visualization

Figure 6 provides a detailed visualization of spiking activity, attention maps, and local field potentials throughout the convergence process. To better illustrate cell assemblies, neuron indices along the

Table 13: Part, Whole, and Nest Scores of the same Composer on scenes with varying numbers of objects (mean $\pm$ std over 5 seeds, 1000 scenes per condition).

Dataset	# objects	Part	Whole	Nest
SHOPs	2	$0.678_{\pm.009}$	$0.914_{\pm.005}$	$0.906_{\pm.005}$
	3	$0.672_{\pm.002}$	$0.867_{\pm.004}$	$0.824_{\pm.002}$
	4	$0.517_{\pm.005}$	$0.775_{\pm.005}$	$0.763_{\pm.004}$
	5	$0.358_{\pm.005}$	$0.636_{\pm.004}$	$0.602_{\pm.005}$
Squares	2	$0.601_{\pm.004}$	$0.947_{\pm.001}$	$0.683_{\pm.002}$
	3	$0.677_{\pm.002}$	$0.953_{\pm.001}$	$0.604_{\pm.002}$
	4	$0.402_{\pm.006}$	$0.852_{\pm.003}$	$0.514_{\pm.007}$
	5	$0.251_{\pm.006}$	$0.629_{\pm.006}$	$0.250_{\pm.007}$
Ts	2	$0.642_{\pm.005}$	$0.639_{\pm.006}$	$0.689_{\pm.004}$
	4	$0.754_{\pm.003}$	$0.715_{\pm.004}$	$0.655_{\pm.007}$
	6	$0.720_{\pm.006}$	$0.694_{\pm.004}$	$0.566_{\pm.006}$
	8	$0.612_{\pm.004}$	$0.642_{\pm.005}$	$0.432_{\pm.005}$
	10	$0.513_{\pm.002}$	$0.523_{\pm.004}$	$0.256_{\pm.001}$
Double-Digit MNIST	2	$0.165_{\pm.003}$	$0.379_{\pm.008}$	$0.436_{\pm.005}$
	3	$0.060_{\pm.003}$	$0.318_{\pm.004}$	$0.231_{\pm.007}$
	4	$-0.105_{\pm.002}$	$0.019_{\pm.003}$	$0.080_{\pm.001}$

y-axis are reordered such that neurons encoding the same object appear adjacent (Fig.6c). Spikes are color-coded based on the ground-truth assignment of each neuron in the SCS (i.e., its corresponding pixel in the input image), allowing each assembly's represented object to be inferred directly by its color. This mapping is verifiable by comparing the circled assemblies in the spike raster (Fig.6c), the zoomed-in spiking pattern (Fig.6d), and the ground-truth objects (Fig.6a). These visualizations show that synchronized cell assemblies emerge gradually from initial randomness. Each assembly corresponds to a part or whole, and assemblies across levels are coordinated in accordance with the part-whole structure.

A notable observation is the reorganization of assemblies during simulation: from phase II to phase III (Fig. 6c), the two green-colored part assemblies reverse their relative order. This reflects the multi-stability of the part-whole solution in Composer, highlighting its dynamic system nature. Such flexibility is essential for representing uncertainty and alternative interpretations of hierarchical structure—an ability often lacking in conventional feedforward or purely supervised networks [6].

Figure 6e shows that top-down attention signals also evolve into structured patterns. Neuron indices are reordered consistently with the spike raster, and attention intensities are visualized via color depth. Notably, the attention pattern mirrors the spiking structure but evolves more slowly, indicating that attention helps regulate spike timing. This modulation is not unidirectional: rather, it reflects iterative interplay between bottom-up spike integration and top-down modulation. Both the DAE and SCS components are therefore essential to achieving consistent hierarchical parsing.

Finally, Fig. 6 also shows the emergence of oscillatory activity at the part level, as measured by the local field potential (LFP), defined as the sum of top-down feedback: $LFP_1(t) = \sum_{i=1}^{N} \gamma_{1i}(t)$, where i indexes neurons in the part-level SCS.

A.8.4 Additional visualizations

Figure 7 presents representative visualizations for each dataset. For more comprehensive examples, Figures 22 to 29 provide two cases per dataset: one successful example and one failure case, illustrating the model's variability in performance across conditions.

A.8.5 Multi-stability in part-whole inference

Figure 1 in the main text highlights a core challenge in part-whole representation: the same visual scene can support multiple structurally consistent interpretations. To demonstrate that Composer exhibits this multi-stability, we introduce a new dataset—the Tetris dataset—where each whole object is composed of four square parts arranged in one of 15 possible configurations (Fig. 30).

We follow the training protocol described in Sec.A.5, and evaluate the model on an ambiguous input image containing two overlapping Tetris objects (Fig.30, top left). The same input yields two distinct interpretations (middle and right), reflecting alternative part-whole groupings.

To visualize the inferred structure, each square in the input is assigned a unique color (bottom left), which is then used to color-code the spiking activity of the corresponding neurons. This is consistent with the visualization strategy used in the main text. However, since no ground truth exists for whole-level grouping in this ambiguous case, coloring is used purely for illustrative purposes.

Notably, we observe that Composer produces two qualitatively different but internally coherent hierarchical organizations, each corresponding to a plausible interpretation of the input. This result illustrates the model's ability to perform content-sensitive, self-consistent structure inference in the absence of unambiguous supervision.

A.8.6 Three-level hierarchy

As discussed in the main text, the Composer framework can in principle be extended beyond two levels, since both columnar modules and cross-level pathways are modular and reusable. Here, we demonstrate such a generalization by implementing a three-level Composer on the SHOPs dataset, where the entire image is treated as an additional top-level "scene" node.

Figure 31 illustrates the emergence of hierarchically nested cell assemblies and a corresponding structured attention sequence. At the scene level, no switching is required, and the assemblies exhibit slow, collective oscillations. The top-level attention signal remains sustained over time. For visualization, scene-level neurons and spikes are color-coded using whole-level ground-truth labels for simplicity.

This result provides empirical support that the Composer architecture scales naturally to deeper hierarchies, enabling parsing at multiple abstraction levels.

A.8.7 Encoding RGB images

While the dataset used in the main text consists of binary images, we show in this subsection that the proposed framework can be naturally extended to handle RGB-valued visual inputs, consistent with how binary spiking neurons encode real-valued images [105–108].

The core idea is conceptually aligned with patch-based encoding, where each pixel or local image patch is transformed into a population vector representation of length L_c, enabling richer representations than a single scalar unit. Accordingly, each pixel (or patch) is represented as a binary vector (Fig. 32a,b), and the denoising autoencoder (DAE) is trained to reconstruct the full population representation of size $W \times H \times L_c$. Fig. 32c,d illustrate that neuronal assemblies corresponding to part–whole representations of the input image (Fig. 32a) emerge naturally in the extended Composer.

In our illustrative example (Fig. 32), we use one-hot binary vectors to encode three discretized RGB values (red, green, and blue). In general, the encoding dimension L_c determines the resolution of color quantization, with larger values corresponding to finer-grained representations of color space.

For low-precision settings, one-hot encoding provides a simple and efficient implementation. For higher-fidelity scenarios, such as natural image modeling, denser binary embeddings can be used, where each RGB value is mapped to a distributed binary code and similarity in color space is reflected in the overlap between codes.

Moreover, biological evidence suggests that color space is non-uniformly represented in neural systems, with certain chromatic directions recruiting larger neural populations than others. In addition, color tuning may exhibit a patch-like organization [105]. This suggests a potential opportunity to design non-uniform or adaptive encoding schemes that allocate representational resources efficiently according to perceptual or biological priors. Finally, neuromorphic hardware [109] is promising to scale the spike-based coding and computation.

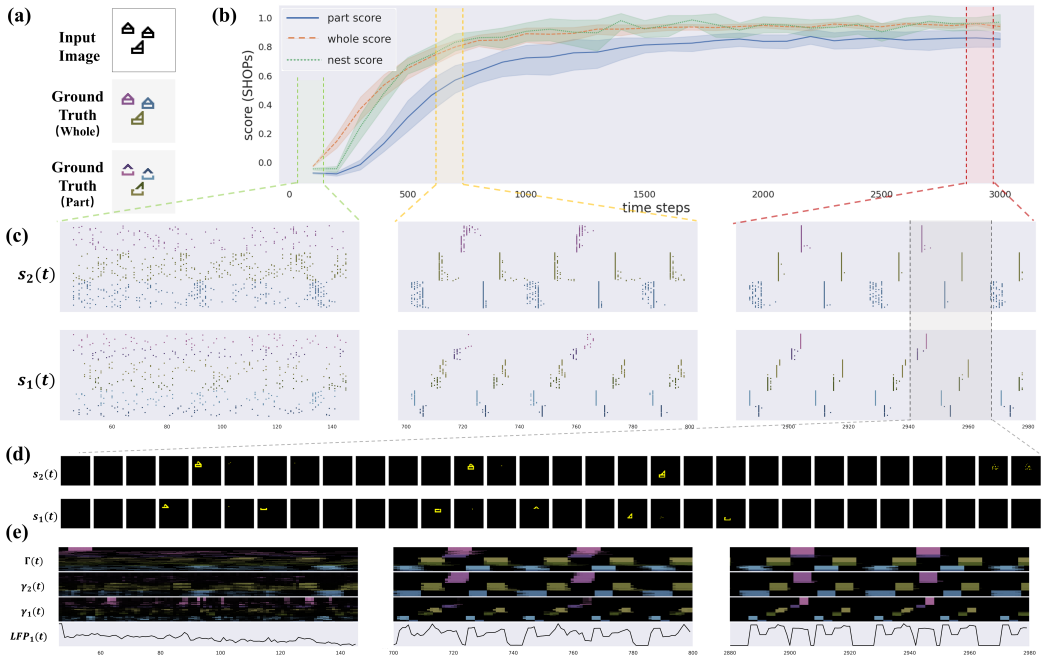

Figure 22: Additional visualization on SHOPs dataset.

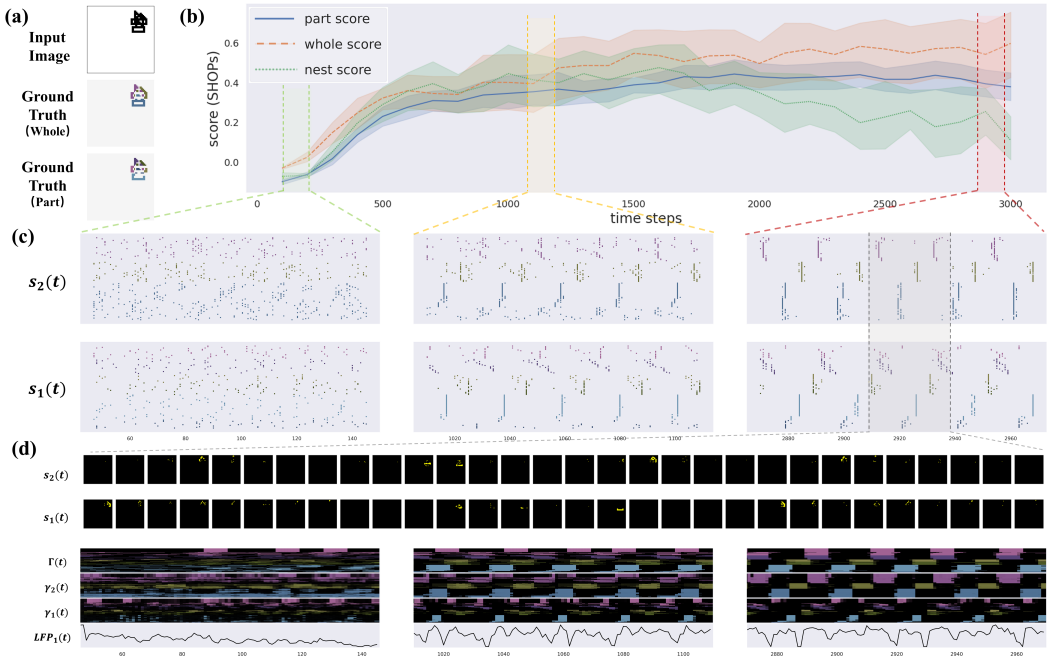

Figure 23: Additional visualization on SHOPs dataset: The fail case, when objects heavily overlap.

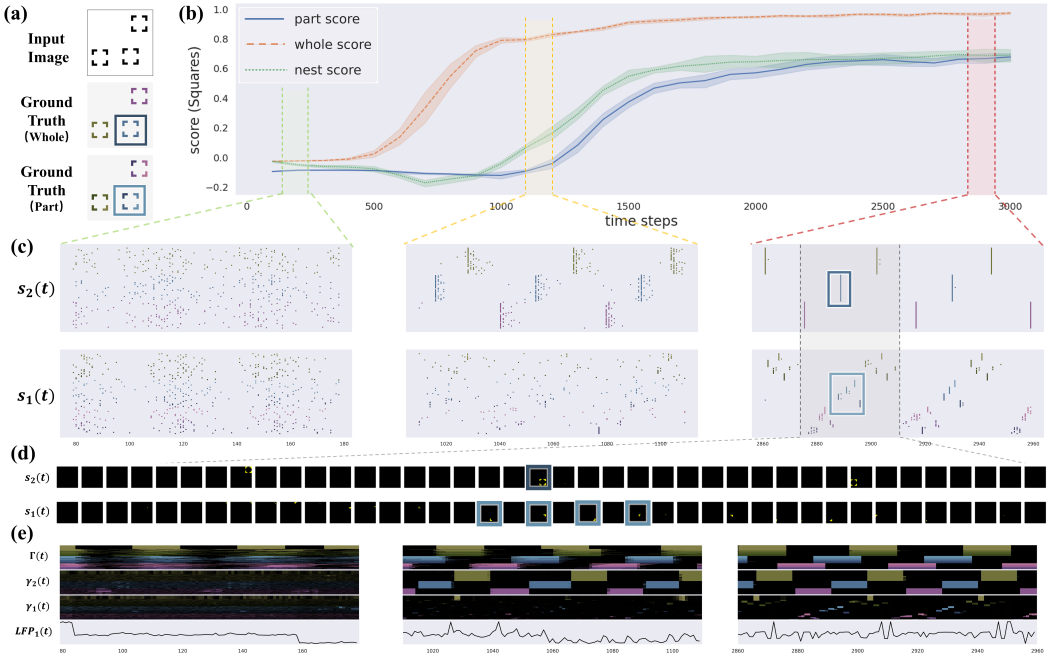

Figure 24: Additional visualization on Squares dataset: Squares do not overlap.

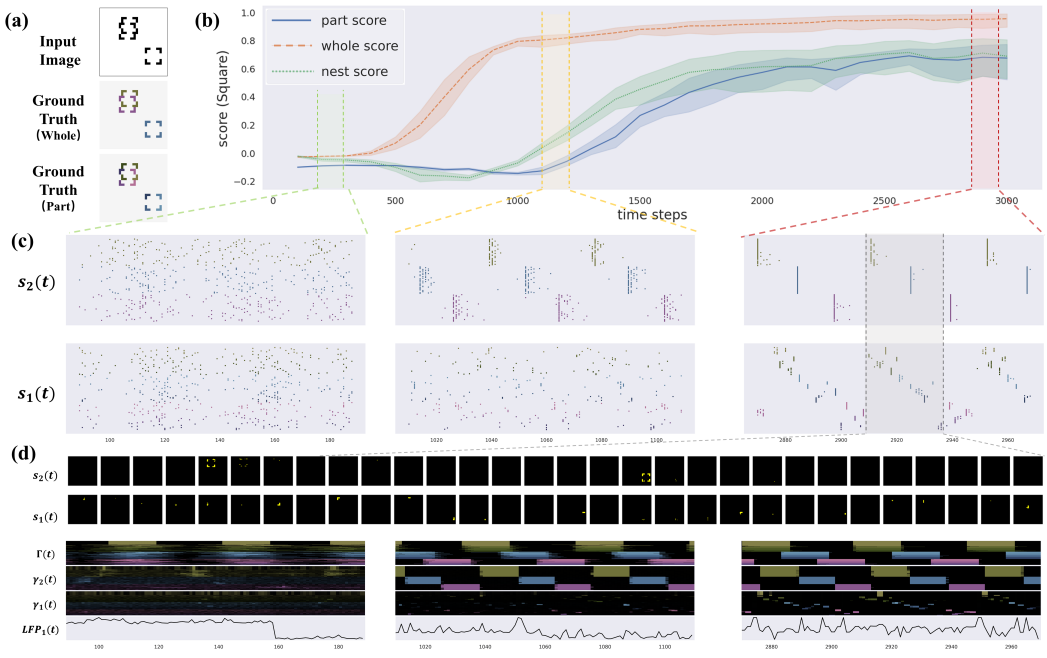

Figure 25: Additional visualization on Squares dataset: Two Squares heavily overlap but the nested structure remains.

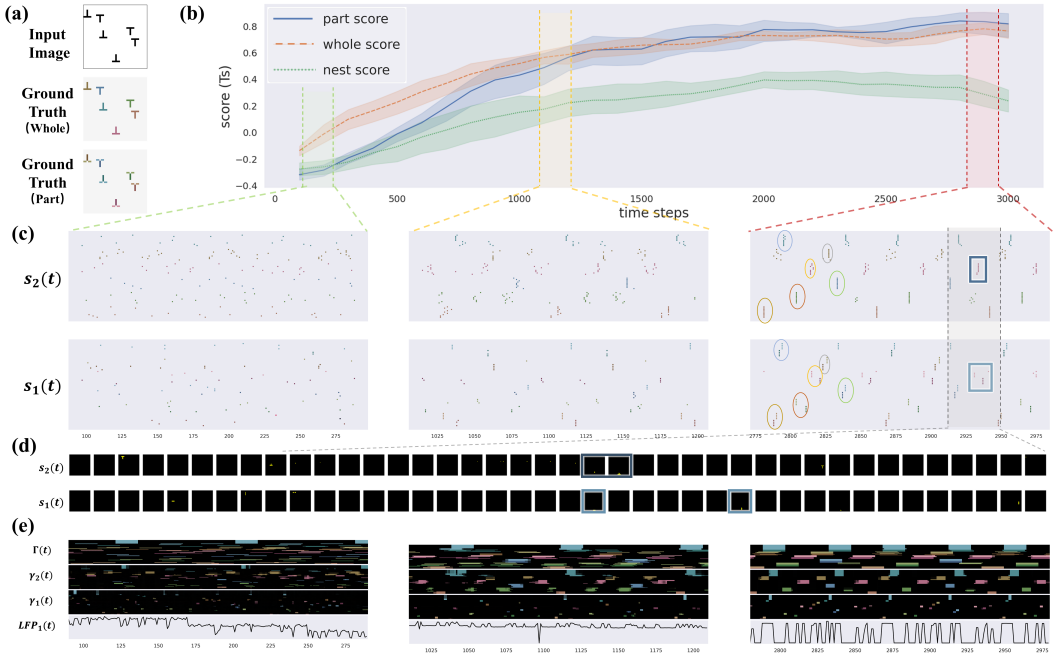

Figure 26: Additional visualization on Ts dataset. Colored circles indicate the coordinated cell assemblies. Same color indicates the part-whole relationship.

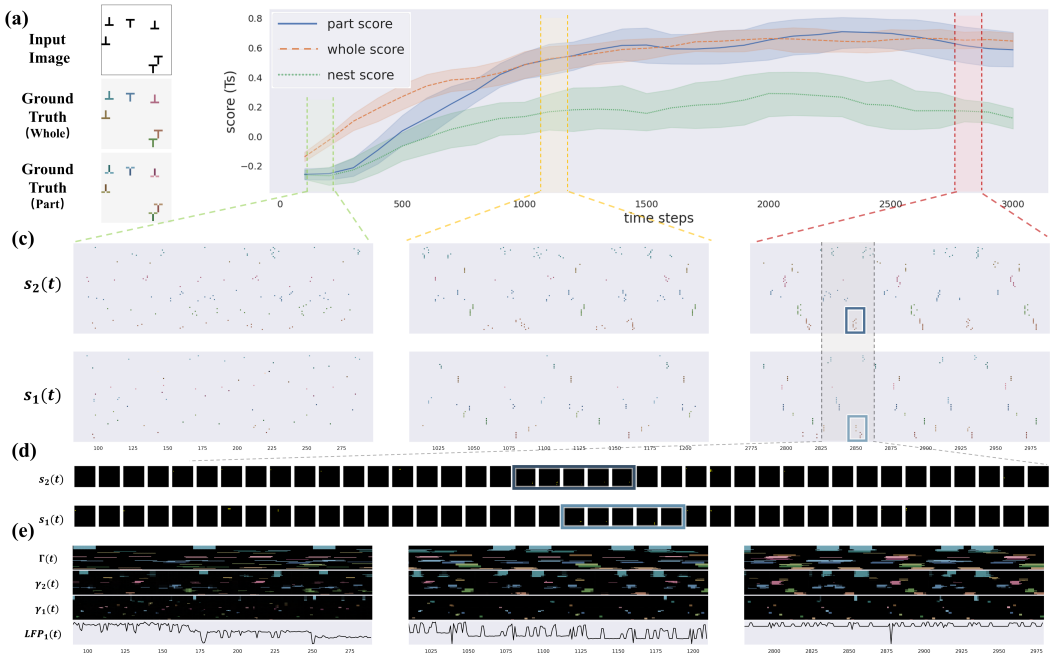

Figure 27: Additional visualization on Ts dataset. Nestedness is not as clear as Fig.26.

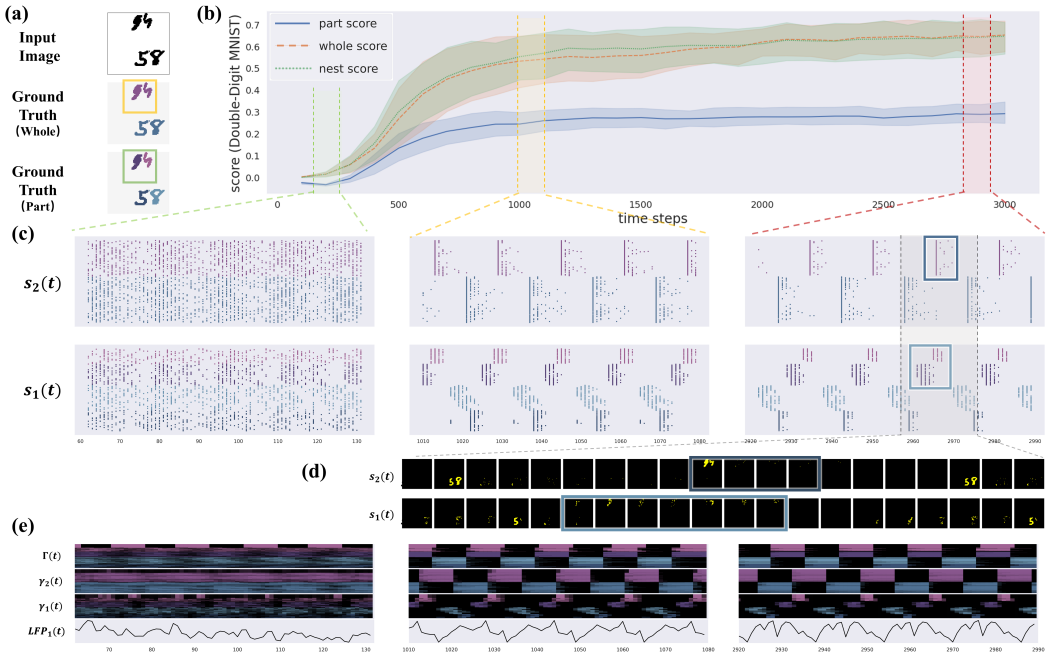

Figure 28: Additional visualization on Double-Digit-MNIST dataset.

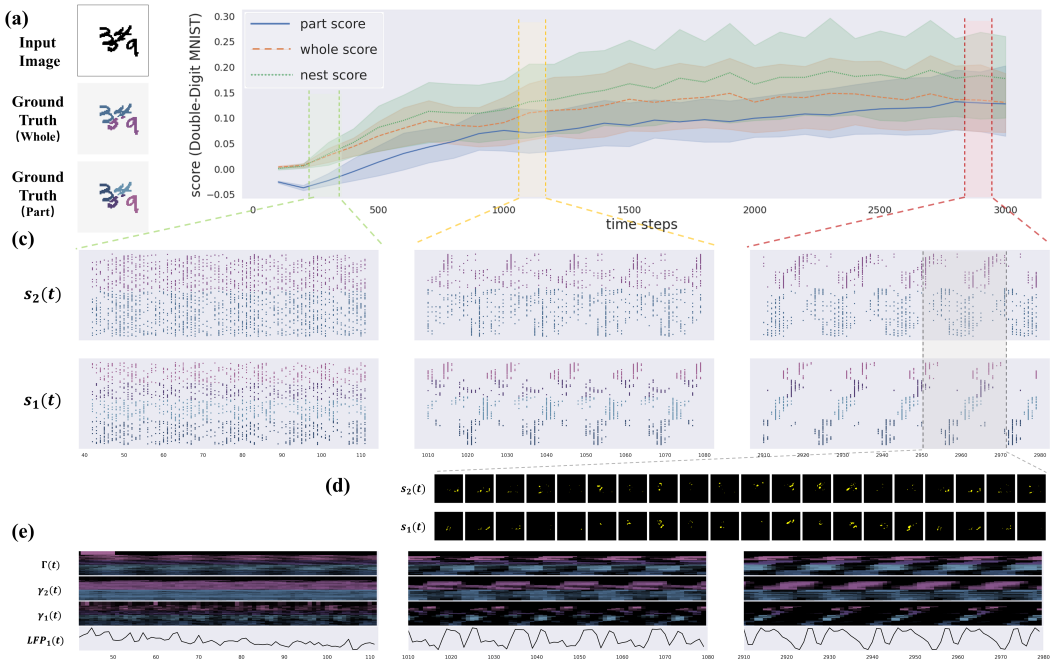

Figure 29: Additional visualization on Double-Digit-MNIST dataset: Digits are crowded and the coherence structure is weaker.

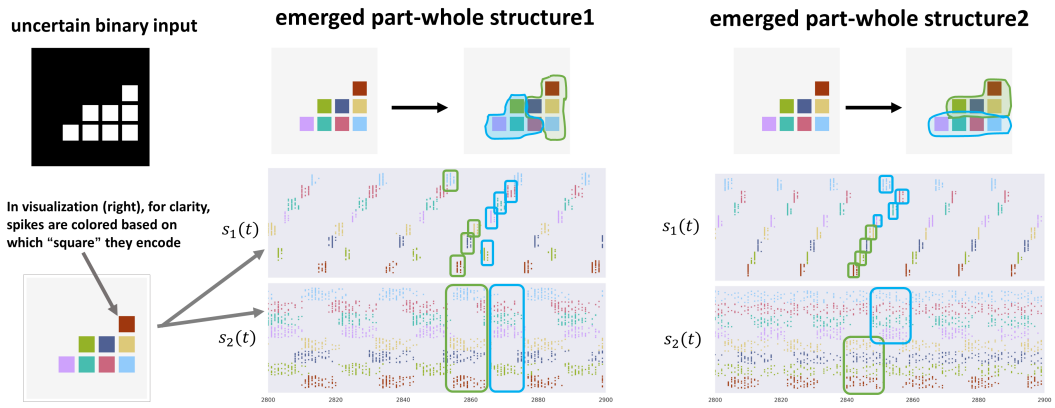

Figure 30: The Composer can infer different reasonable part-whole structures when the input is "ambiguous" in terms of compositional structure. We train the model in a new Tetris dataset. Part objects are squares, and whole objects are composed of 4 squares in various ways (15 possibilities). Then when it tries to decompose the ambiguous input, multiple part-whole structures emerge as mentioned in Fig.1 in main paper.

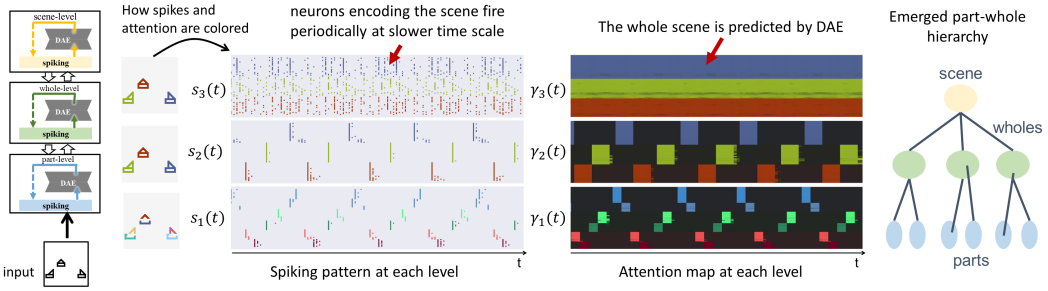

Figure 31: Extending the Composer to 3 levels to account for scene-level, the top-level neurons fire coherently at slow time scale ($s_3(t)$).

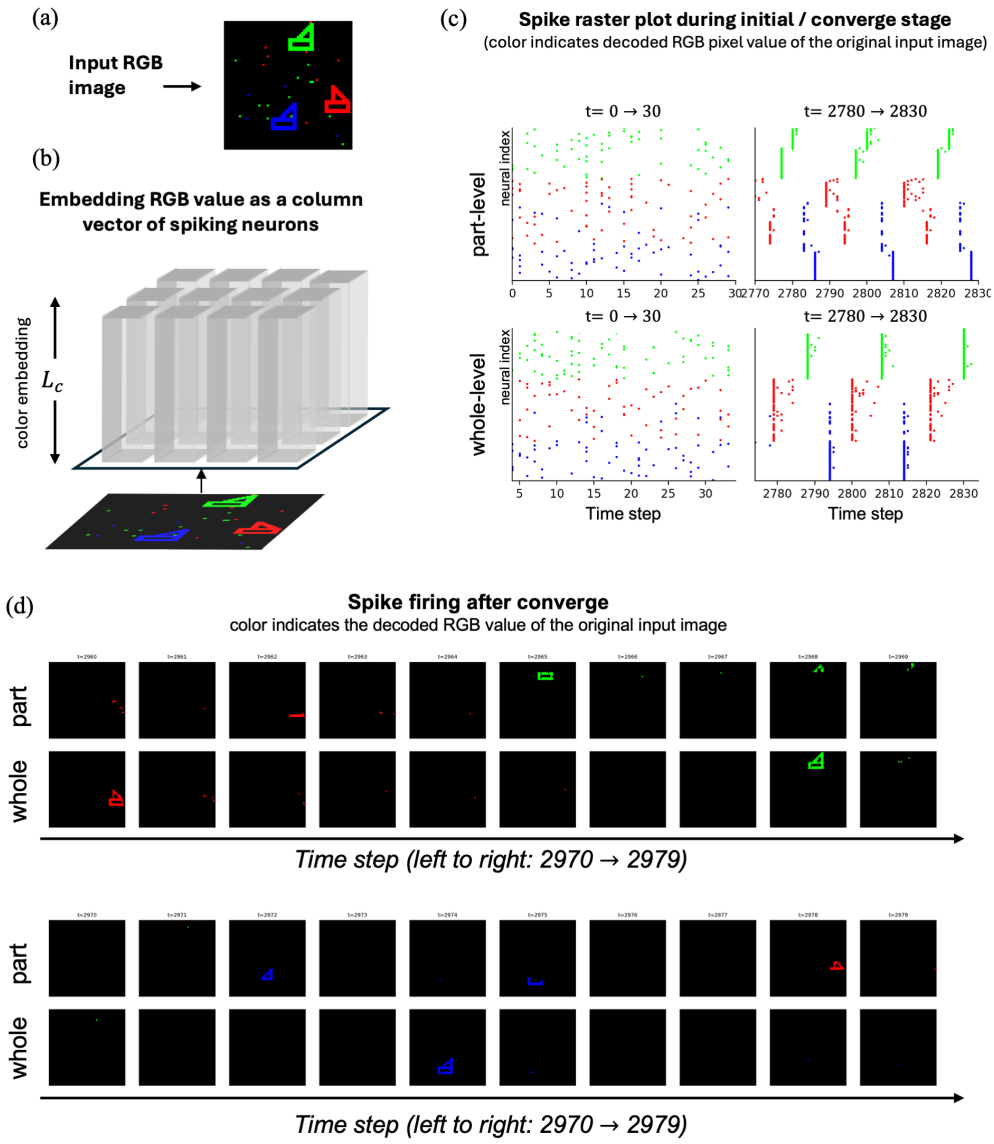

Figure 32: Extending the Composer to handle RGB images. (a) Example input image, adapted from the SHOPs dataset. (b) Real-valued RGB inputs are encoded as column binary vectors, using either a one-hot or a dense encoding scheme. This formulation allows the entire pipeline to be naturally extended to color inputs. Notably, this encoding strategy is consistent with color coding principles observed in biological visual systems [105–108]. (c) Spike raster plots, where the y-axis denotes neuron indices. Each spike is colored according to the RGB value of the corresponding input pixel. The left panel shows the initial phase, and the right panel shows the converged phase. (d) Time-resolved firing dynamics of the model during converged phase. Spikes are colored according to the RGB values of the corresponding input image.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The contribution of this work is clearly clarified in the abstract and introduction, including three parts: representation hypothesis, model implementation and structure-aware evaluation towards the problem of representing part-whole relationship.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of the proposed model and its applicability are discussed in detail in Appendix A.1, including the simplicity of the current datasets, architectural assumptions, simplified computation and learning schemes, and the need for further evaluation on real-world data.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: For each theoretical result (e.g., Theorem 1), we explicitly state all assumptions and provide a complete and correct proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide a complete description of the model architecture (Appendix A.4), training pipeline (Appendix A.5), and evaluation procedure (Appendix A.3) in the Appendix. We also provide code to reproduce all main experimental results (Appendix A.2).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to both the code and datasets required to reproduce the main experimental results, along with detailed instructions (See Appendix A.2).

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide a self-contained description of the experimental setup in the main text, augmented with additional details on data splits, training configurations, and evaluation procedures in Appendix A.5.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report error bars based on evaluations over 5 random seeds. The variance is consistently low across datasets, indicating stable average performance. Details are shown in Table 7 and Table 8.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide details on hardware specifications in Appendix A.5.1

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research does not involve human subjects, sensitive data, or applications with foreseeable ethical concerns. All experiments and data generation conform to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This paper is motivated by fundamental questions in AI, psychology, and cognitive neuroscience, and is not tied to a specific downstream application. As such, the current work does not have an immediate or direct societal impact. However, the proposed ideas could influence future applications depending on how they are integrated into practical systems. In a general sense, representation learning methods may be incorporated into downstream tasks such as perception, decision-making, or automation, where unintended biases or misinterpretations could arise. At the same time, the proposed approach emphasizes interpretability and alignment with biological principles, which may facilitate more transparent and controllable system behavior. We believe that careful evaluation and responsible integration will be important for ensuring positive societal outcomes in future applications. These considerations are broadly applicable to neural-network-based methods, and are thus not further elaborated in this work.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our work does not involve models or datasets with known high-risk misuse potential. We use only synthetic datasets and do not release generative models or internet-scraped content.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in this paper are publicly available and properly cited. We use MNIST under its public license and re-implemented the Agglomerator following its open-source repository and citation [17]. No assets requiring special access or restricted terms were used.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release new synthetic datasets and model implementations (Composer). All assets are accompanied by documentation and configuration files to support reproducibility. Details are provided in the Appendix (Sec. A.2).

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing or any research with human participants.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects and thus does not require IRB approval.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as part of the core methodology or in any way that affects the scientific contributions of this work. Any language assistance was limited to writing or editing support.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.