POSS: LEARNING REUSABLE STATE EVIDENCE FOR COMPOSITIONAL ZERO-SHOT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Compositional zero-shot learning (CZSL) requires visual–semantic knowledge that can be reused in novel state–object combinations. Yet distinguishing seen composition labels need not produce such knowledge: competition between labels can discourage state evidence that remains visually valid. We study this mismatch through the semantic scope of supervision. A composition label specifies some properties of an object, while its classification objective compares it with candidates describing other properties. We introduce Partially Observed State Supervision (POSS), which organizes existing labels into sparse semantic slot–value facts and learns object-conditioned visual evidence only from the facts each annotation reveals. Shared state predictors pool supervision across training compositions, and their evidence contributes to the ranking of both seen and unseen candidates. POSS retains standard single-composition prediction and requires no additional image-level training annotations. Experiments on three benchmarks and multiple CZSL architectures show improvements in closed- and open-world recognition. State-retention diagnostics show improved preservation of visually supported states, while transfer analyses link reusable state evidence to gains on unseen compositions.

1 INTRODUCTION

Compositional zero-shot learning (CZSL) recognizes unseen combinations of familiar states and objects. In fixed-category classification with similar training and test distributions, features tailored to the annotated class distinctions can support accurate prediction. CZSL places an additional demand on those features: evidence for a familiar state must remain useful when its object pairing changes. Its central challenge is therefore to learn visual–semantic knowledge that transfers beyond the observed combinations.

Existing methods improve compositional representations through factorization, cross-modal interaction, contextual modeling, and structured reasoning (Nayak et al., 2023; Lu et al., 2023; Huang et al., 2024; Wang et al., 2023; Wu et al., 2025b). Work on invariant representations also studies correlations that favor seen pairs (Zhang et al., 2022). We examine a specific source of difficulty in the supervision itself: *the competition needed to choose a composition label can extend beyond the state information that the label actually supplies*.

Consider the schematic example of a brown, wet dog. If its annotation is *wet dog*, a composition classifier is trained to rank that label above *brown dog*. This ranking is appropriate for the benchmark's single-label target, but it does not establish that the image lacks brown coloration. When the same competitive signal directly shapes state predictors, it can suppress valid visual evidence merely because a different property was selected for annotation. Such interference ties state responses to the training set's annotation preferences, leaving weaker evidence for unseen compositions that require those same properties.

Our objective is therefore to improve recognition under the standard single-label CZSL protocol, rather than to reformulate CZSL as a multi-label task. Each training image retains its original composition label, and evaluation still requires one predicted composition. State learning serves this same decision by providing evidence that can be reused across pairings.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

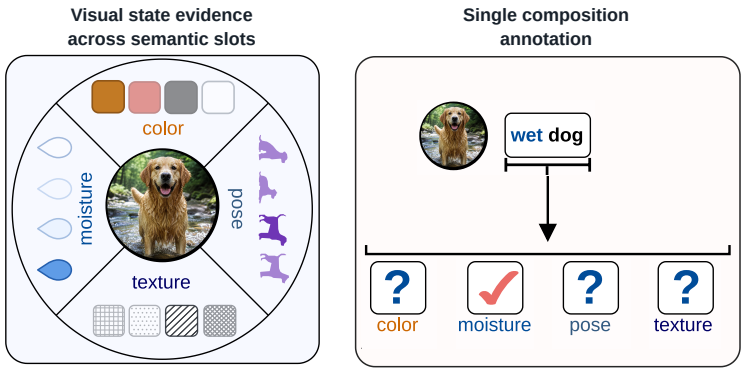


Figure 1: A single composition label reveals only part of the image's state evidence. The label *wet dog* supervises moisture; question marks denote unspecified supervision in other slots. Ranking the annotated composition above alternatives is a classification requirement. Suppressing their visual evidence instead ties state learning to annotation preferences and hinders reuse in new combinations.

We introduce *Partially Observed State Supervision* (POSS), which uses each label's semantic scope to organize state learning. It maps state labels to sparse sets of semantic slot–value facts, learns object-conditioned evidence for those facts, and uses that evidence to score candidate compositions. Supervision from different training compositions updates shared state predictors, allowing a new pairing to reuse an already learned visual–semantic correspondence. A composition scorer retains the pair-level distinctions needed for the final single-label decision. The framework uses existing training annotations and keeps the standard CZSL candidate spaces and prediction rule.

Our contributions are threefold:

- We identify and investigate a mismatch between composition discrimination and reusable state learning: label competition can impose state-level constraints unsupported by the annotation's semantic scope.
- We introduce POSS, a framework that connects label semantics, object-conditioned state learning, and composition scoring through reusable state evidence.
- Experiments on three benchmarks show consistent gains across multiple CZSL architectures. POSS (Troika) achieves the highest AUC and harmonic mean among the compared methods under both closed- and open-world evaluation. Ablations and diagnostics show the benefits of preserving state evidence and using it in composition scoring.

2 Related Work

Representations for compositional generalization. Semantic embeddings and learned similarity functions support transfer to new categories (Akata et al., 2016; Sung et al., 2018). In CZSL, hierarchical decomposition, attribute transformations, and graph embeddings construct representations of unseen pairs (Yang et al., 2020; Li et al., 2020; Naeem et al., 2021). Disentanglement and attention methods separate state and object features (Saini et al., 2022; Hao et al., 2023). Invariant feature learning aligns representations and gradients across primitive-defined domains to reduce dependence on seen-pair correlations (Zhang et al., 2022). POSS examines the supervision that shapes these representations: separating features into branches still leaves open which state constraints a composition label supports. Semantic slots make this scope explicit and provide an inductive structure for learning interpretable factors (Locatello et al., 2019).

Vision–language models for CZSL. CSP learns primitive-specific prompt tokens that can be recombined for unseen pairs (Nayak et al., 2023). DFSP fuses decomposed language features with image features (Lu et al., 2023), while Troika jointly models states, objects, and compositions and adapts prompt representations to visual content (Huang et al., 2024). PLID models diverse class contexts through language-informed distributions (Bao et al., 2024), and ClusPro represents each

primitive with multiple clustering-based prototypes (Qu et al., 2025). Conditional and logic-based approaches model dependencies between primitives and compositions (Wang et al., 2023; Wu et al., 2025a;b). POSS adds object-conditioned, reference-corrected state evidence to these composition scorers, allowing the same supervision design to accompany different architectures.

Learning from incomplete annotations. Partially supervised CZSL addresses examples missing the state or object annotation (Karthik et al., 2022; Huo et al., 2024; Panda & Mukherjee, 2026). KG-SP uses external feasibility knowledge to help estimate missing primitive labels, while ProCC learns cross-primitive compatibility through progressive training. Multi-label learning with missing annotations studies incomplete label matrices (Yu et al., 2014); single-positive learning estimates unobserved labels or constrains the expected number of positives (Cole et al., 2021). Class-aware selective loss uses class frequencies and sample-specific likelihoods to handle unannotated entries (Ben-Baruch et al., 2022), and MIME refines pseudo-labels through mutual label enhancement (Liu et al., 2023). POSS starts from the standard annotated state–object pair and derives the semantic scope of its supervision. Shared predictors pool the facts supplied by different training images, providing state evidence for unseen pairings. The objective remains recognition of one composition under the original single-label CZSL protocol.

3 Composition Supervision and Reusable State Knowledge

Task. Training examples (x, a, o) contain an image and an annotated state–object pair from the seen set $\mathcal{C}_{\mathrm{s}}$. Unseen pairs $\mathcal{C}_{\mathrm{u}}$ contain familiar primitives in new combinations. Evaluation considers $\mathcal{C}_{\mathrm{s}} \cup \mathcal{C}_{\mathrm{u}}$ in closed world and $\mathcal{A} \times \mathcal{O}$ in open world. The target in both cases is one exact composition.

Two roles of a composition label. The annotation identifies the classification target and supplies evidence about a limited set of object properties. These roles support different constraints. Choosing *wet dog* over *brown dog* specifies their required ranking, while learning moisture from *wet* supplies no color target. A state predictor directly exposed to all composition competitors can therefore receive updates unrelated to the property currently supervised. We view this as a source of interference in reusable state learning. State responses can become tied to the training set's annotation preferences, weakening evidence for properties that remain visually present. This leaves the model with less reliable state knowledge when those properties are needed to recognize an unseen composition.

Learning evidence that can be reused. POSS organizes supervision by semantic slots such as moisture or material. A state label determines the slots whose values can be learned from that example; predictors are shared across examples and conditioned on the object. At test time, each candidate selects the state facts relevant to its own meaning. A state response learned from one set of compositions thereby supplies visual evidence for new candidates that share the same semantic facts. Preserving this evidence gives unseen pairings access to knowledge learned from the training images, connecting state supervision directly to single-label composition recognition.

4 Partially Observed State Supervision

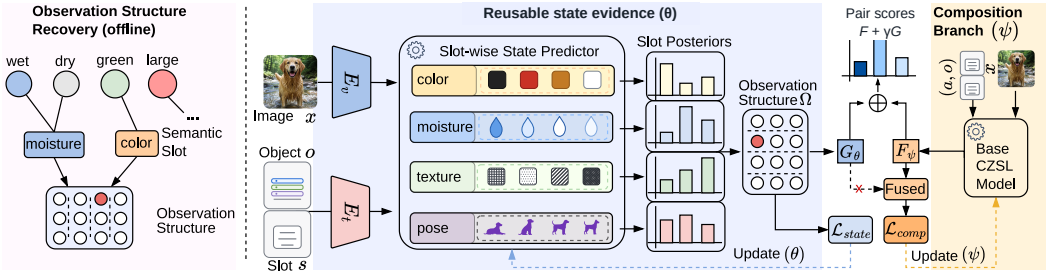
POSS connects three components: semantic observation structures specify supervision, object-conditioned predictors learn visual state evidence, and candidate-specific evidence contributes to composition scoring (Figure 2).

4.1 Semantic Scope of State Supervision

Let $\mathcal{S}$ denote semantic slots and $\mathcal{V}_s$ the value vocabulary for slot s. For each state primitive a, we recover a sparse mapping

$$\Omega(a) \subseteq \bigcup_{s \in \mathcal{S}} \{s\} \times \mathcal{V}_s, \qquad m(a) = |\Omega(a)|. \tag{1}$$

An entry (s, v) means that the label specifies value v of slot s; for example, $\Omega(\textit{wet}) = \{(\texttt{moisture}, \texttt{wet})\}$. Each mapping contains at most one value per slot. Unmentioned slots receive no state target from that example.



Train: annotated facts Ω(a). Score: candidate facts Ω(a); reference-centered evidence G = mean log(q / $\bar{q}$).

Figure 2: POSS learns reusable state evidence and applies it to composition recognition. A vocabulary-level structure Ω identifies the facts expressed by each state label. Object-conditioned predictors learn the corresponding visual evidence from shared image features. During training, the annotated label selects the supervised facts; during inference, each candidate selects its own facts for evidence scoring. The reference-centered state score G adjusts the composition score F through $F + \gamma G$, which determines a single predicted pair. The shared visual adapter is omitted from branch labels for clarity; the training objective is specified in Eq. (7).

We recover this vocabulary-level structure once using GPT-5.5 and Gemini 2.5 Pro (OpenAI, 2026; Comanici et al., 2025). Independent inventory proposals are reconciled into a canonical slot–value vocabulary; each model then maps primitives into it, and agreed assignments are retained. Material labels additionally retain their material-identity atom, as specified by the UT-Zappos recovery protocol. The procedure uses training-split vocabularies, with no images, predictions, test compositions, or evaluation feedback. The structures are shared across training runs. For uncovered labels, $\Omega(a) = \varnothing$ and the state contribution is zero.

4.2 Learning Object-Conditioned State Evidence

Frozen CLIP visual and text encoders (Radford et al., 2021), together with a trainable visual adapter ϕ (Chen et al., 2022), produce image tokens H_x and a normalized object embedding e_o. A slot query extracts visual evidence conditional on the candidate object:

$$\begin{aligned} r_s(o) &= u_s + B_s e_o, \\ f_s(x, o) &= \mathrm{norm}(A_s[\mathrm{Attn}(r_s(o), H_x, H_x); r_s(o)]) \,. \end{aligned} \tag{2}$$

Here $\theta_s = \{u_s, B_s, A_s\}$ contains the slot-specific parameters, Attn is scaled dot-product attention, and brackets denote concatenation. Fixed text prompts describe object o with slot value v, giving normalized prototypes $t_{s,v,o}$. The within-slot prediction is

$$q_{\theta,\phi,s}(v \mid x, o) = \frac{\exp(f_s(x, o)^\top t_{s,v,o}/\tau_s)}{\sum_{v' \in \mathcal{V}_s} \exp(f_s(x, o)^\top t_{s,v',o}/\tau_s)}. \tag{3}$$

Object conditioning allows a shared state vocabulary to retain differences in appearance across objects.

For a mini-batch $\mathcal{B}$, let $M_\mathcal{B} = \sum_{i \in \mathcal{B}} m(a_i)$. The state objective uses only the facts expressed by each training label:

$$\mathcal{L}_{\text{state}} = -\frac{1}{M_\mathcal{B}} \sum_{i \in \mathcal{B}} \sum_{(s,v) \in \Omega(a_i)} \log q_{\theta,\phi,s}(v \mid x_i, o_i). \tag{4}$$

It is zero when $M_\mathcal{B} = 0$. Under a factorized slot decoder, this is the negative log-likelihood of the mapped state facts, averaged over the observed facts in the batch; unmentioned slots are marginalized out. Each slot is trained on the examples that reveal it, and different images update the same predictor, pooling supervision across seen compositions.

4.3 Composition Scoring with State Evidence

A candidate (a, o) queries the predictors for its own facts $\Omega(a)$. Their evidence is measured relative to a fixed, add-one-smoothed training frequency $\bar{q}_s(v)$:

$$G_{\theta,\phi}(x, a, o) = \frac{1}{m(a)} \sum_{(s,v)\in\Omega(a)} \log \frac{q_{\theta,\phi,s}(v \mid x, o)}{\bar{q}_s(v)}, \tag{5}$$

with $G = 0$ for $m(a) = 0$. To compute $\bar{q}_s(v)$, we count training images whose mapped labels reveal (s, v), add one to every value's count, and normalize within each slot. The reference is fixed before optimization. Averaging limits dependence on mapping size, while referencing measures support relative to each state's training frequency. Raw $\log q$ would give every covered candidate a nonpositive contribution while uncovered candidates receive zero.

Let $F_{\psi,\phi}(x, a, o)$ be a base CZSL scorer. Prediction combines its pair-level discrimination with the corresponding state evidence:

$$\widehat{c} = \arg \max_{(a,o)\in\mathcal{C}_{\text{eval}}} \left[F_{\psi,\phi}(x, a, o) + \gamma G_{\theta,\phi}(x, a, o)\right]. \tag{6}$$

In probability space, this reweights the base posterior by $\exp(\gamma G)$ and normalizes over candidate compositions. The state score thus adjusts candidate margins using visual support for their mapped facts. Shared predictors supply this evidence for unseen pairs as well as seen ones, allowing state supervision on the training images to guide recognition of new combinations.

4.4 Training

We train with the exact composition label and the partial-state objective:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}\big(F + \gamma\, \text{sg}[G], c\big) + \lambda \mathcal{L}_{\text{state}}. \tag{7}$$

Cross-entropy normalizes over $\mathcal{C}_{\text{s}}$ and is averaged over the batch. The stop-gradient operator sg retains the state score in the forward prediction while preventing the composition loss from directly updating θ through G. The shared adapter ϕ receives gradients through F and $\mathcal{L}_{\text{state}}$, coupling the visual features to both objectives. This training rule retains composition discrimination while learning the state-specific parameters from the semantic facts expressed by the labels. After training, Eq. (6) selects the highest-scoring candidate composition.

Derivations, structure-recovery protocols, and full implementation settings are provided in Appendices A, C, and D.

5 Experiments

5.1 Experimental Setup

We use the standard MIT-States, UT-Zappos, and C-GQA splits (Isola et al., 2015; Yu & Grauman, 2014; Naeem et al., 2021). Each training image retains its original composition label. Observation structures are recovered from training vocabularies and fixed across runs, except in the structure and coverage experiments.

POSS denotes our model with an adapter-enhanced CoOp-style composition scorer that learns text prompts (Zhou et al., 2022). POSS (DFSP) and POSS (Troika) use the corresponding architecture for $F_{\psi,\phi}$ while retaining the same state-learning and evidence-scoring design (Lu et al., 2023; Huang et al., 2024). All POSS variants freeze CLIP ViT-L/14 and train the visual adapter, prompts, and POSS modules. We use AdamW for at most 30 epochs, select checkpoints by validation AUC, and report means over seeds $\{0, 1, 2\}$. Default weights are $\lambda = 1.0$ and $\gamma = 0.5$. Unless specified, component studies use the CoOp-based POSS on MIT-States with the consensus structure.

Closed-world (CW) evaluation considers the benchmark's seen and unseen compositions; open-world (OW) evaluation considers all state–object pairs. Both require one composition prediction per image. We sweep the seen–unseen calibration bias to obtain the best seen accuracy (S), unseen accuracy (U), harmonic mean (HM), and area under the seen–unseen curve (AUC). Main-text comparisons report AUC/HM under the same splits, candidate spaces, and calibration protocol.

Table 1: Standard CZSL results. AUC and HM are reported for closed-world (CW) and open-world (OW) evaluation; best values among the listed methods are bold. POSS entries are means over three seeds.

	MIT-States		UT-Zappos		C-GQA	
Method	AUC	HM	AUC	HM	AUC	HM
Closed world						
CLIP (2021)	11.0	26.1	5.0	15.6	1.4	8.6
CoOp (2022)	13.5	29.8	18.8	34.6	4.4	17.1
DFSP (2023)	20.6	37.3	36.0	47.2	10.5	27.1
Troika (2024)	22.1	39.3	41.7	54.6	12.4	29.4
ClusPro (2025)	23.8	40.7	46.6	58.5	14.9	32.8
Logic-Troika (2025b)	23.4	40.5	45.8	57.8	15.3	33.3
I2CD (2026)	23.3	40.0	46.4	57.9	16.3	35.0
EVA (2026)	24.0	41.0	50.2	60.2	18.8	36.9
FlowComposer (2026)	23.5	40.2	46.8	58.6	15.9	34.0
POSS	24.6	41.7	46.3	58.5	19.2	37.3
POSS (DFSP)	25.0	42.1	47.0	59.4	19.4	37.5
POSS (Troika)	**25.8**	**42.8**	**50.4**	**61.9**	**19.9**	**38.2**
Open world						
CLIP (2021)	3.0	12.8	2.2	11.2	0.3	4.0
CoOp (2022)	2.8	12.3	13.2	28.9	0.7	5.5
DFSP (2023)	6.8	19.3	30.3	44.0	2.4	10.4
Troika (2024)	7.2	20.1	33.0	47.8	2.7	10.9
ClusPro (2025)	9.3	23.0	39.5	54.1	3.0	11.6
Logic-Troika (2025b)	8.7	22.4	36.2	50.8	3.4	12.6
I2CD (2026)	8.2	21.4	35.6	50.2	4.3	14.9
EVA (2026)	9.4	22.8	40.2	54.2	5.6	17.9
FlowComposer (2026)	7.5	20.3	35.5	51.0	3.5	12.6
POSS	9.6	23.1	37.9	52.7	5.7	17.8
POSS (DFSP)	9.9	24.0	38.9	53.8	5.9	18.2
POSS (Troika)	**10.3**	**24.2**	**40.8**	**55.5**	**6.1**	**18.8**

Table 2: Ablations on MIT-States (open world). The first group contains reference controls; each row in the middle group changes one component of full POSS.

Variant	Control or change	HM	AUC
CoOp	Composition-only reference	12.3	2.8
+ MLP	Add matched capacity without state supervision	12.8	2.9
Flat aux.	Predict the original flat state-label vocabulary	15.7	4.6
Dense-neg.	Add negative targets for unobserved slot values	18.0	6.0
Global feat.	Replace slot attention with a global image feature	20.9	8.1
No-Obj	Remove object conditioning from the state branch	21.6	8.6
No-SG	Allow composition gradients into the state branch	20.5	7.8
POSS	Full model	**23.1**	**9.6**

5.2 Recognition of Seen and Unseen Compositions

Table 1 includes CLIP, CoOp, and specialized CZSL methods. The DFSP/POSS (DFSP) and Troika/POSS (Troika) comparisons test whether state evidence helps different composition architectures. POSS improves both scorers in every dataset–protocol comparison. POSS (Troika) achieves the highest AUC and HM among the compared methods in all six settings. The following experiments examine where these gains come from: learning useful state evidence, preserving it during training, and using it to rank candidates.

Table 3: Observation structures on MIT-States (OW). Cov.: covered primitives (%); Prec.: atom precision (%) from independent human verification.

Structure	Cov.	Prec.	HM	AUC
No structure	–	–	12.8	2.9
Single slot	100.0	–	17.7	5.3
Random	96.5	19.2	16.9	5.0
Human	87.8	96.5	22.8	9.4
GPT-5.5	93.9	95.7	22.8	9.5
Gemini 2.5 Pro	91.3	94.1	22.6	9.3
Consensus	87.0	97.4	**23.1**	**9.6**

Table 4: State recognition on MIT-States. Observed: annotated slots; Unobserved: other verified visible states. Accuracies are slot-wise macro-averages.

Variant	Observed	Unobserved	
	Acc. ↑	Acc. ↑	NLL ↓
Flat aux.	55.1	24.0	2.91
Dense-neg.	61.3	19.8	3.24
No-SG	62.1	30.3	2.54
POSS	**63.5**	**36.4**	**2.08**

5.3 Learning Useful State Evidence

Table 2 uses CoOp as the composition-only reference. + MLP adds a capacity-matched multilayer perceptron trained only for composition recognition. Flat aux. trains an auxiliary classifier on the original state labels without slot grouping. Dense-neg. retains POSS's slot branch but treats values in unmentioned slots as negative targets. The gap to POSS shows that extra capacity and conventional state supervision do not account for the full gain.

The remaining variants change one component: Global feat. replaces slot attention with a shared global image feature; No-Obj removes object conditioning from the state branch; No-SG lets the composition loss update state parameters without changing forward scores. These results support object-conditioned, slot-specific predictors trained on the facts revealed by each label.

Table 3 compares structure controls and recovery sources. No structure reuses + MLP; Single slot merges all state primitives into one POSS slot; Random assigns one to three valid inventory atoms per covered primitive without semantic associations. Mappings from humans and each recovery model are compared with Consensus, which retains the two models' agreed assignments. Semantic mappings outperform the controls, and consensus achieves the highest precision and AUC despite lower coverage.

5.4 Preservation of Visually Supported States

Table 4 evaluates the trained state predictors on 612 held-out MIT-States images with 938 additionally verified state labels. These extra labels are used only for diagnosis. For each verified slot, we predict its value and macro-average the metrics across slots. Observed refers to the slot named by the original annotation; Unobserved refers to other visible states omitted from that annotation. Dense-neg. reaches 61.3% observed-slot accuracy but only 19.8% on other visible states; POSS reaches 63.5% and 36.4%, respectively. Thus, good recognition of the annotated state can coexist with poor preservation of other visual evidence.

We also measure how updates change already learned predictions. Each variant restarts from the same trained checkpoint; we freeze the composition scorer and update only the state branch for 100 steps using matched data counts and optimizer settings. For updates on another slot of the same object, we measure closed-world accuracy on held-out target-slot images relative to its step-0 value. Retention is 95.0% for Dense-neg., 96.3% for No-SG, and 99.5% for POSS. This controlled comparison shows that POSS better preserves existing state evidence as training moves across semantic dimensions.

5.5 Using State Evidence for Composition Recognition

Table 5(a) uses full POSS as the reference and switches off one contribution at a time. Setting $\lambda = 0$ keeps evidence fusion but removes direct state supervision; setting $\gamma = 0$ keeps state supervision but uses only the composition score for prediction. The latter still trains the shared visual adapter with the state loss, isolating what explicit evidence fusion adds beyond auxiliary representation learning.

Table 5: State evidence in composition recognition on MIT-States. Entries are means over three seeds. Panel (a) uses the empirical reference; panel (b) fixes $\lambda = 1.0$ and $\gamma = 0.5$.

	Closed world		Open world	
Setting	AUC	HM	AUC	HM
(a) State supervision and evidence fusion				
No state supervision ($\lambda = 0, \gamma = 0.5$)	16.4	34.0	4.9	16.1
No evidence fusion ($\lambda = 1.0, \gamma = 0$)	17.3	34.8	4.6	16.0
Full POSS ($\lambda = 1.0, \gamma = 0.5$)	**24.6**	**41.7**	**9.6**	**23.1**
(b) Evidence reference				
Uncentered	12.7	28.8	3.1	12.7
Uncentered + learned offset	18.6	36.0	6.4	18.5
Uniform	22.9	40.3	8.4	21.8
Max centering	23.7	41.1	9.0	22.4
Empirical (default)	**24.6**	**41.7**	**9.6**	**23.1**

Table 6: Dependence on semantic coverage on MIT-States. A fixed fraction of the 108 retained slot–value facts is dropped uniformly; entries are means over three training seeds.

		Closed world		Open world	
Dropped	Retained facts	AUC	HM	AUC	HM
0%	108	24.6	41.7	9.6	23.1
25%	81	22.8	40.2	8.5	21.7
50%	54	20.3	38.0	7.0	19.8
75%	27	17.8	35.4	5.8	17.8
100%	0	14.1	31.8	2.9	12.8

Open-world AUC falls from 9.6 to 4.9 and 4.6, respectively. Learning state evidence and using it in candidate scores are both needed for the full gain.

Panel (b) compares raw $\log q$ (Uncentered), raw evidence with one learned offset shared by covered candidates, equal within-slot references (Uniform), each input's largest within-slot probability (Max centering), and smoothed training frequencies (Empirical). These choices affect both training logits and final scores. Raw evidence favors uncovered candidates, yielding 3.1 open-world AUC; a global offset raises this to 6.4, while the empirical reference reaches 9.6. This supports value-specific references beyond a global score correction.

5.6 Relation to Unseen-Composition Gains

Table 6 tests how much semantic coverage POSS needs. We uniformly drop facts from the 108-fact MIT-States mapping, removing them from both state supervision and candidate scoring. Each mask is fixed across three training seeds; the composition scorer and remaining training settings are retained. Open-world AUC decreases from 9.6 with all facts to 7.0 with half and 2.9 with none. Even 27 facts give 5.8 AUC. Partial coverage is therefore useful, while broader coverage extends supervision and evidence to more candidates.

Figure 3 compares exact unseen-composition accuracy against CoOp + Adapter, the matched composition scorer without POSS. Cross-slot exposure measures the fraction of an object's covered training images whose labels describe only slots outside the target state's slots. Support counts training images carrying each mapped fact, averaged for multi-fact labels. Gains grow with cross-slot exposure, consistent with the proposed interference mechanism. States with adequate training support also benefit more. These trends connect unseen-composition gains to the availability and preservation of reusable state evidence.

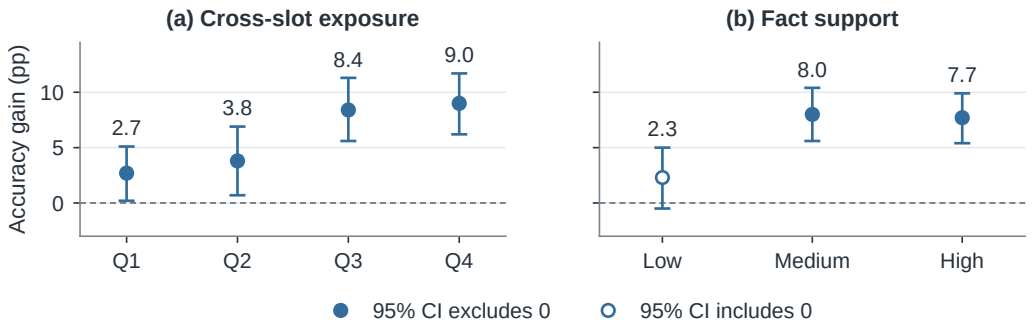


Figure 3: Unseen-composition gains over CoOp + Adapter on MIT-States (open world), with 95% composition-bootstrap intervals. Exposure increases from Q1 to Q4.

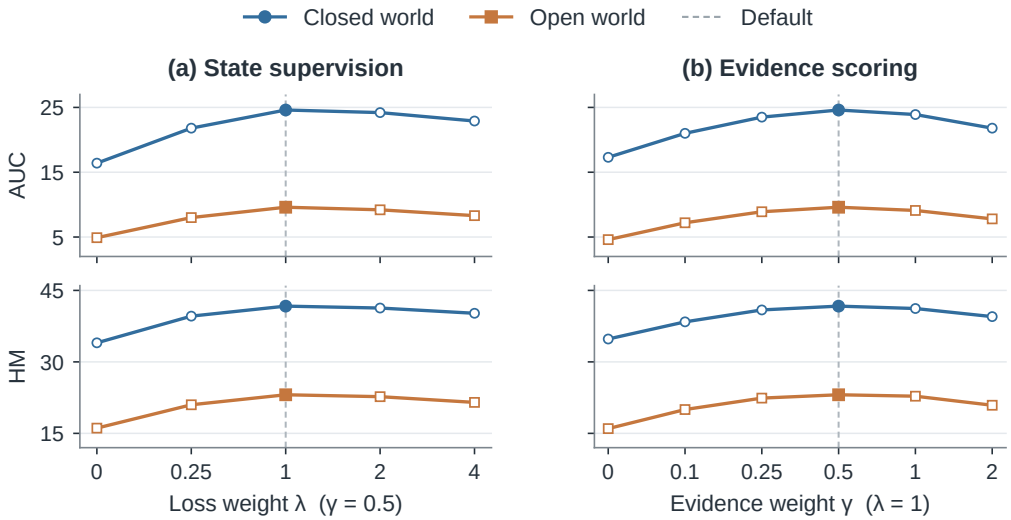


Figure 4: Sensitivity to λ and γ on MIT-States (three-seed means). Ticks enumerate tested settings at equal spacing; dashed lines mark the defaults.

5.7 Hyperparameter Sensitivity

Figure 4 varies one coefficient at a time, holding the other at its default. Both sweeps peak at $(\lambda, \gamma) = (1.0, 0.5)$. Nearby settings remain strong: $\lambda = 2.0$ gives 9.2 open-world AUC, while $\gamma = 0.25$ and 1.0 give 8.9 and 9.1, versus 9.6 at the default. Larger weights ($\lambda = 4.0$ or $\gamma = 2.0$) reduce AUC to 8.3 or 7.8. Moderate weights thus balance state supervision with composition learning and state evidence with the composition score.

Appendices B–F provide complete metrics and variability, structure audits, implementation details, computational cost, diagnostic protocols, and qualitative examples.

6 Conclusion

POSS connects composition supervision with state knowledge that transfers to new pairings. By organizing existing labels by their semantic scope, it learns object-conditioned state evidence and uses this evidence to guide exact-composition decisions. Consistent improvements across benchmarks and architectures, together with state-preservation diagnostics and scoring ablations, support this approach to compositional generalization. These results highlight supervision design as an effective complement to compositional representation learning. Limitations and practical considerations are discussed in Appendix G.

AI USE STATEMENT

Generative AI tools were used in a supporting role throughout the preparation of this work. Their use included improving the wording, organization, and presentation of the manuscript; assisting with translation and LaTeX editing; supporting literature retrieval and related-work summarization; and providing limited help with code revision and debugging during experimentation.

A separate use of generative AI was part of the POSS pipeline itself. GPT-5.5 and Gemini 2.5 Pro were used to construct the vocabulary-level observation structures through the consensus-based recovery procedure described in Section 4.1 and Appendix C. This procedure operated on the training vocabulary rather than on experimental outcomes or evaluation feedback.

AI-generated suggestions were treated as intermediate material rather than final research outputs. References suggested by AI tools were checked against the original sources, code changes were reviewed and tested before use, and manuscript descriptions were cross-checked against the implemented methods and experimental settings. The reported results were produced by the authors' own implementations and experiments, and the research questions, methodological choices, experimental design, and scientific interpretation remained under the authors' control.

The authors take responsibility for the accuracy of the manuscript, the validity of its claims, and all reported experimental results.

REFERENCES

Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7): 1425–1438, 2016.

Wentao Bao, Lichang Chen, Heng Huang, and Yu Kong. Prompting language-informed distribution for compositional zero-shot learning. In *European Conference on Computer Vision*, pp. 107–123. Springer, 2024.

Emanuel Ben-Baruch, Tal Ridnik, Itamar Friedman, Avi Ben-Cohen, Nadav Zamir, Asaf Noy, and Lihi Zelnik-Manor. Multi-label classification with partial annotations using class-aware selective loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4764–4772, 2022.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

Elijah Cole, Oisin Mac Aodha, Titouan Lorieul, Pietro Perona, Dan Morris, and Nebojsa Jojic. Multi-label learning from single positive labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 933–942, 2021.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Shaozhe Hao, Kai Han, and Kwan-Yee K. Wong. Learning attention as disentangler for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15315–15324, June 2023.

Zhenqi He, Lin Li, and Long Chen. FlowComposer: Composable flows for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Siteng Huang, Biao Gong, Yutong Feng, Min Zhang, Yiliang Lv, and Donglin Wang. Troika: Multi-path cross-modal traction for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24005–24014, 2024.

Fushuo Huo, Wenchao Xu, Song Guo, Jingcai Guo, Haozhao Wang, Ziming Liu, and Xiaocheng Lu. ProCC: Progressive cross-primitive compatibility for open-world compositional zero-shot learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12689–12697, 2024.

Phillip Isola, Joseph J Lim, and Edward H Adelson. Discovering states and transformations in image collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1383–1391, 2015.

Shyamgopal Karthik, Massimiliano Mancini, and Zeynep Akata. KG-SP: Knowledge guided simple primitives for open world compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9336–9345, 2022.

Yong-Lu Li, Yue Xu, Xiaohan Mao, and Cewu Lu. Symmetry and group in attribute-object compositions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11316–11325, 2020.

Biao Liu, Ning Xu, Jiaqi Lv, and Xin Geng. Revisiting pseudo-label for single-positive multi-label learning. In *International Conference on Machine Learning*, pp. 22249–22265. PMLR, 2023.

Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pp. 4114–4124. PMLR, 2019.

Xiaocheng Lu, Song Guo, Ziming Liu, and Jingcai Guo. Decomposed soft prompt guided fusion enhancing for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23560–23569, 2023.

Muhammad Ferjad Naeem, Yongqin Xian, Federico Tombari, and Zeynep Akata. Learning graph embeddings for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 953–962, 2021.

Nihal V Nayak, Peilin Yu, and Stephen H Bach. Learning to compose soft prompts for compositional zero-shot learning. In *International Conference on Learning Representations*, 2023.

OpenAI. GPT-5.5 System Card. OpenAI, 2026. URL `https://openai.com/index/gpt-5-5-system-card/`. Accessed: July 2026.

Aditya Panda and Dipti Prasad Mukherjee. Partially supervised compositional zero-shot learning by class-balanced distribution alignment. *IEEE Transactions on Image Processing*, 35:2484–2498, 2026. doi: 10.1109/TIP.2026.3664762.

Hongyu Qu, Jianan Wei, Xiangbo Shu, and Wenguan Wang. Learning clustering-based prototypes for compositional zero-shot learning. In *International Conference on Learning Representations*, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Nirat Saini, Khoi Pham, and Abhinav Shrivastava. Disentangling visual embeddings for attributes and objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13658–13667, 2022.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1199–1208, 2018.

Qingsheng Wang, Lingqiao Liu, Chenchen Jing, Hao Chen, Guoqiang Liang, Peng Wang, and Chunhua Shen. Learning conditional attributes for compositional zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11197–11206, 2023.

Peng Wu, Qiuxia Lai, Hao Fang, Guo-Sen Xie, Yilong Yin, Xiankai Lu, and Wenguan Wang. A conditional probability framework for compositional zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3673–3683, 2025a.

Peng Wu, Xiankai Lu, Hao Hu, Yongqin Xian, Jianbing Shen, and Wenguan Wang. LOGICZSL: Exploring logic-induced representation for compositional zero-shot learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30301–30311, 2025b.

Muli Yang, Cheng Deng, Junchi Yan, Xianglong Liu, and Dacheng Tao. Learning unseen concepts via hierarchical decomposition and composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10248–10256, 2020.

Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 192–199, 2014.

Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. Large-scale multi-label learning with missing labels. In *International Conference on Machine Learning*, pp. 593–601. PMLR, 2014.

Zhaoquan Yuan, Zining Wang, Yuankang Pan, Ao Luo, Wei Li, Xiao Wu, and Changsheng Xu. I2CD: An invertible causal framework for compositional zero-shot learning via disentangle-compose-disentangle. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12295–12303, 2026.

Tian Zhang, Kongming Liang, Ruoyi Du, Xian Sun, Zhanyu Ma, and Jun Guo. Learning invariant visual representations for compositional zero-shot learning. In *European Conference on Computer Vision*, 2022. URL https://arxiv.org/abs/2206.00415.

Xiao Zhang, Haodong Jing, Yongqiang Ma, and Nanning Zheng. Decoupling primitive with experts: Dynamic feature alignment for compositional zero-shot learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

APPENDIX

A NOTES ON THE TRAINING OBJECTIVE

We give the key identities behind the partial-state loss and evidence score. Parameter subscripts are omitted where they do not affect the argument. The likelihood interpretation treats the retained facts in $\Omega(a)$ as valid observations; recovery quality is evaluated in Appendix C.

A.1 LIKELIHOOD OF THE OBSERVED FACTS

Let $S(a)$ contain the slots revealed by label a, with values $v_{S(a)}$. For a factorized decoder $q(Z \mid x, o) = \prod_s q_s(Z_s \mid x, o)$, summing over the unmentioned slots gives

$$q\big(Z_{S(a)} = v_{S(a)} \mid x, o\big) = \prod_{(s,v)\in\Omega(a)} q_s(v \mid x, o). \tag{8}$$

Each unmentioned slot contributes $\sum_v q_s(v \mid x, o) = 1$ to this marginalization. Taking the negative logarithm and averaging over observed facts yields Eq. (4). An empty mapping contributes zero loss. Factorization specifies the decoder; it does not require physical properties to be independent in the data.

Observation scope. Let R_s indicate whether the annotation reveals slot s. For the ideal per-slot risk $\mathbb{E}[-R_s \log q_s(Z_s \mid X, O)]$, conditioning on an input with positive observation probability gives cross-entropy against $p(Z_s \mid X, O, R_s = 1)$. This is the population target of that risk. It equals the full slot posterior when $R_s \perp Z_s \mid X, O$; otherwise, supervision reflects the subpopulation in which the slot is revealed.

A.2 HOW STATE EVIDENCE CHANGES CANDIDATE SCORES

Let $\pi_F(c \mid x) = \mathrm{softmax}_c\, F(x, c)$ over the evaluation candidates. The fused score in Eq. (6) gives

$$\widetilde{\pi}(c \mid x) = \frac{\pi_F(c \mid x)\exp(\gamma G(x, c))}{\sum_{c'} \pi_F(c' \mid x)\exp(\gamma G(x, c'))} = \mathrm{softmax}_c(F(x, c) + \gamma G(x, c))\,. \tag{9}$$

This follows by substituting the softmax expression for π_F and canceling its common normalizer. For two candidates, their log-odds change by $\gamma[G(x, c_1) - G(x, c_2)]$: the state evidence adjusts the margin used for the same single-composition decision. Setting $\gamma = 0$ recovers π_F.

Raw $\log q_s(v \mid x, o)$ is nonpositive, whereas an uncovered candidate receives zero state score. Using raw evidence would therefore systematically penalize covered candidates relative to uncovered ones. The reference $\bar{q}_s(v)$ in Eq. (5) measures support relative to the value's observed training frequency. The fixed reference enters candidate scores and the fused training logits; the partial-state likelihood remains unchanged.

A.3 WHICH PREDICTORS RECEIVE SUPERVISION

For a coupled composition score $\widetilde{F}(x, c')$ with softmax probability $\pi(c' \mid x)$ and annotated target c, cross-entropy gives

$$\frac{\partial \ell_{\mathrm{comp}}}{\partial \widetilde{F}(x, c')} = \pi(c' \mid x) - \mathbb{I}[c' = c]. \tag{10}$$

A competing composition receives a downward training signal even when it names a slot unobserved in the current annotation. When that score depends on a state predictor, the signal can directly update its parameters.

In POSS, a slot-specific parameter block θ_t receives a direct state-loss gradient from an example only when $t \in S(a)$; an example that does not reveal t contains no loss term for that predictor. The training rule in Eq. (7) also blocks the direct composition-loss path through G. The shared visual adapter ϕ still receives gradients from both the composition scorer F and the state loss. These update paths explain the scope of the gradient audit in Appendix F.1.

Table 7: Full MIT-States results for the methods in Table 1. CW and OW denote closed and open world.

	CW				OW			
Method	AUC	HM	S	U	AUC	HM	S	U
CLIP	11.0	26.1	30.2	46.0	3.0	12.8	30.1	14.3
CoOp	13.5	29.8	34.4	47.6	2.8	12.3	34.6	9.3
DFSP	20.6	37.3	46.9	52.0	6.8	19.3	47.5	18.5
Troika	22.1	39.3	49.0	53.0	7.2	20.1	48.8	18.7
ClusPro	23.8	40.7	52.1	54.0	9.3	23.0	51.2	22.1
Logic-Troika	23.4	40.5	50.8	53.9	8.7	22.4	50.7	21.4
I2CD	23.3	40.0	50.6	54.1	8.2	21.4	50.6	20.6
EVA	24.0	41.0	51.2	55.0	9.4	22.8	50.8	23.2
FlowComposer	23.5	40.2	51.5	53.2	7.5	20.3	50.4	19.0
POSS	24.6	41.7	51.6	56.1	9.6	23.1	51.5	23.7
POSS (DFSP)	25.0	42.1	52.2	56.0	9.9	24.0	51.8	23.8
POSS (Troika)	**25.8**	**42.8**	53.6	55.0	**10.3**	**24.2**	53.4	23.0

Table 8: Full UT-Zappos results for the methods in Table 1. CW and OW denote closed and open world.

	CW				OW			
Method	AUC	HM	S	U	AUC	HM	S	U
CLIP	5.0	15.6	15.8	49.1	2.2	11.2	15.7	20.6
CoOp	18.8	34.6	52.1	49.3	13.2	28.9	52.1	31.5
DFSP	36.0	47.2	66.7	71.7	30.3	44.0	66.8	60.0
Troika	41.7	54.6	66.8	73.8	33.0	47.8	66.4	61.2
ClusPro	46.6	58.5	70.7	76.0	39.5	54.1	71.0	66.2
Logic-Troika	45.8	57.8	69.6	74.9	36.2	50.8	69.6	63.7
I2CD	46.4	57.9	69.1	77.1	35.6	50.2	69.1	62.1
EVA	50.2	60.2	71.2	79.6	40.2	54.2	71.6	66.5
FlowComposer	46.8	58.6	71.1	74.9	35.5	51.0	70.1	61.2
POSS	46.3	58.5	70.5	78.9	37.9	52.7	70.3	66.5
POSS (DFSP)	47.0	59.4	72.6	79.4	38.9	53.8	71.8	67.3
POSS (Troika)	**50.4**	**61.9**	74.7	80.7	**40.8**	**55.5**	72.7	68.2

B Complete Composition-Recognition Results

Tables 7–9 retain all four metrics from the original comparison. Each metric is evaluated under the standard seen–unseen calibration sweep; S and U need not occur at the HM operating point.

Table 9: Full C-GQA results for the methods in Table 1. CW and OW denote closed and open world.

	CW				OW			
Method	AUC	HM	S	U	AUC	HM	S	U
CLIP	1.4	8.6	7.5	25.0	0.3	4.0	7.5	4.6
CoOp	4.4	17.1	20.5	26.8	0.7	5.5	21.0	4.6
DFSP	10.5	27.1	38.2	32.0	2.4	10.4	38.3	7.2
Troika	12.4	29.4	41.0	35.7	2.7	10.9	40.8	7.9
ClusPro	14.9	32.8	44.3	37.8	3.0	11.6	41.6	8.3
Logic-Troika	15.3	33.3	44.4	39.4	3.4	12.6	43.7	9.3
I2CD	16.3	35.0	44.1	41.2	4.3	14.9	44.0	10.9
EVA	18.8	36.9	47.1	44.6	5.6	17.9	46.9	13.3
FlowComposer	15.9	34.0	44.8	40.7	3.5	12.6	43.5	10.2
POSS	19.2	37.3	45.8	44.8	5.7	17.8	45.7	13.5
POSS (DFSP)	19.4	37.5	46.4	43.6	5.9	18.2	46.2	12.9
POSS (Troika)	**19.9**	**38.2**	46.6	46.3	**6.1**	**18.8**	46.4	14.1

Algorithm 1 Two-stage consensus recovery of observation structures

Require: Training state vocabulary $\mathcal{A}$, training object vocabulary $\mathcal{O}$, shared prompts, and output schemas

for $r \in \{\mathrm{GPT}, \mathrm{Gemini}\}$ **do**
  Propose and canonicalize inventory $\mathcal{I}^{(r)}$
end for
Match the proposals to obtain shared inventory $\mathcal{I}$
if $\mathcal{A}$ is the UT-Zappos vocabulary **then**
  Add `material_identity` values
end if
Freeze $\mathcal{I}$
for $r \in \{\mathrm{GPT}, \mathrm{Gemini}\}$ **do**
  Map $\mathcal{A}$ to $\Omega^{(r)}$ in validated batches
end for
for each $a \in \mathcal{A}$ **do**
  if $\mathrm{amb}^{(r)}(a)$ or $\mathrm{oos}^{(r)}(a)$ for some r **then**
    $\Omega(a) \leftarrow \varnothing$
  else
    $\Omega(a) \leftarrow \Omega^{(\mathrm{GPT})}(a) \cap \Omega^{(\mathrm{Gemini})}(a)$
  end if
  if $\mathcal{A}$ is the UT-Zappos vocabulary **then**
    Add the material-identity atom of a
  end if
end for
return $\mathcal{I}$ and $\{\Omega(a) : a \in \mathcal{A}\}$

C Observation-Structure Recovery Details

We instantiate the vocabulary-level observation map Ω in two stages: the models first build a shared slot–value inventory, then independently map each state primitive to that frozen inventory. The procedure runs once per dataset before training.

C.1 Recovery Inputs and Access Restrictions

For each training split, GPT-5.5 (OpenAI, 2026) and Gemini 2.5 Pro (Comanici et al., 2025) receive the same state/object vocabularies, instructions, and schemas. They see no images, composition lists, model outputs, training outcomes, or evaluation metrics. Object primitives only disambiguate state terms; the final structure depends on the state primitive: $\Omega : \mathcal{A} \to 2^{\cup_{s \in \mathcal{S}} \{s\} \times \mathcal{V}_s}$. It therefore creates no image-level or object-specific pseudo-labels.

C.2 Consensus Slot–Value Inventory

The two models independently organize the training vocabulary into semantic slots and canonical values. For recovery source $r \in \{\mathrm{GPT}, \mathrm{Gemini}\}$, let

$$\mathcal{I}^{(r)} = \left\{ \left(s, \mathcal{V}_s^{(r)} \right) \right\}_{s \in \mathcal{S}^{(r)}} \tag{11}$$

denote its proposed inventory. Slot membership follows one *co-instantiation test*: two properties are different slots iff one object can hold both simultaneously. Thus size and thickness are separate, whereas wet and dry are values of one moisture slot; size, width, thickness, and height consequently remain distinct. This gives $|\mathcal{S}| = 26$ slots on MIT-States. Values must be mutually exclusive, slots object-independent, and only true synonyms may be merged; context, viewpoint, emotion, activity, brand, and unpopulated slots are excluded.

The inventories are canonicalized and compared deterministically, with no additional model call. Normalization lowercases names, maps spaces and hyphens to underscores, singularizes, and applies a fixed alias table (for example `wetness`→`moisture`, `soiling`→`cleanliness`, `intactness`). Both models must propose a slot, with matching scope or at least 50% value overlap, for it to be retained; values are retained only when both place them in the matched slot. For non-exact matches, we use the overlap coefficient

$$\mathrm{ov}(s, s') = \frac{|\mathcal{V}_s^{(\mathrm{GPT})} \cap \mathcal{V}_{s'}^{(\mathrm{Gemini})}|}{\min\{|\mathcal{V}_s^{(\mathrm{GPT})}|, |\mathcal{V}_{s'}^{(\mathrm{Gemini})}|\}}. \tag{12}$$

We retain pairs with $\mathrm{ov}(s, s') \geq 0.5$ using maximum-weight one-to-one matching, breaking ties by normalized slot name. The resulting consensus inventory is

$$\mathcal{I} = \{(s, \mathcal{V}_s)\}_{s \in \mathcal{S}}. \tag{13}$$

The resulting $\mathcal{I}$ is frozen for all mapping runs, variants, and seeds; downstream results cannot add slots or values. For UT-Zappos it also contains a `material_identity` slot with one canonical value for each training attribute. This preserves the benchmark label without claiming whole-shoe coverage, even when several materials are visible. This is a benchmark-label identity slot: its values are categorical with respect to the annotated composition, not a claim that physical materials cannot co-occur.

C.3 State-to-Atom Mapping

Both models independently map each primitive $a \in \mathcal{A}$ using the fixed inventory $\mathcal{I}$. Source r returns

$$\Omega^{(r)}(a) \subseteq \bigcup_{s \in \mathcal{S}} \{s\} \times \mathcal{V}_s. \tag{14}$$

Each atom (s, v) must be lexically entailed by a; a primitive may have at most one value per slot, and visual consequences, object correlations, and contextual implications are excluded. For example, *wet* maps to $(\texttt{moisture}, \texttt{wet})$. It does not imply color, texture, temperature, or material. An empty mapping is allowed when no assignment is sufficiently clear.

Dataset-specific recovery for UT-Zappos. UT-Zappos attributes mostly denote materials, so every attribute receives a canonical material-identity atom. At most two auxiliary atoms capture stable surface cues; fine-grained identities remain distinct, and no shoe type, color, geometry, brand, co-occurrence, or whole-object coverage is inferred. Broad labels such as *Leather*, *Cotton*, and *Synthetic* therefore receive sparse mappings, with auxiliary atoms kept only under model agreement. This yields full coverage while preserving material labels.

C.4 Conservative Mapping Consensus

Consensus is deterministic. If either model marks a primitive as ambiguous or out of scope, it is dropped. Otherwise the retained structure is the intersection

$$\Omega(a) = \Omega^{(\mathrm{GPT})}(a) \cap \Omega^{(\mathrm{Gemini})}(a), \tag{15}$$

taken over canonicalized (s, v) pairs. Thus an atom survives only when both models assign the same value to the same slot; disagreements retain neither.

Table 10: Compact prompt contract used for observation-structure recovery. Dataset-specific vocabularies and the frozen inventory are substituted in the indicated input fields; no images, split identities, frequencies, or evaluation results are supplied.

Stage	Supplied content	Required behavior	Exclusions and output
Stage 1: inventory	Dataset name, alphabetized training-state vocabulary, and object vocabulary used only to disambiguate word senses.	Construct the smallest object-independent set of slots. Values within a slot must be mutually exclusive; merge only true synonyms and keep simultaneously applicable dimensions separate.	Exclude scene context, viewpoint, illumination, emotion, activity, subjective judgment, brand, and object-category slots. Return the Stage-1 JSON inventory.
Stage 2: mapping	Dataset name, the frozen inventory, an optional dataset block, and a batch of at most 25 state primitives.	Map each primitive to the minimal set of directly stated slot–value facts. Use only frozen slots and values, allow an empty mapping, and flag ambiguity or out-of-scope meanings.	Exclude visual consequences, correlations, co-occurrence, object identity, and new slots or values. Return at most three atoms per primitive with one-clause justifications.
UT-Zappos override	The frozen `material_identity` inventory and material attributes are supplied through the Stage-2 block.	Include exactly one identity atom and at most two stable cues about the material itself, such as texture, finish, fibers, or weave.	Preserve fine-grained material labels, make no whole-shoe coverage claim, and infer nothing from shoe type, color, shape, gender, brand, price, or frequency. Use the same Stage-2 JSON schema.

Table 11: Inference configuration for observation-structure recovery. Both models receive identical prompts, vocabularies, and output schemas. Stage 2 processes primitives in alphabetical batches of 25 with no cross-batch context, so the token limit applies per batch. A request seed is set wherever the provider's interface supports it.

Setting	GPT-5.5	Gemini 2.5 Pro
Provider	OpenAI	Google
Temperature	0	0
Top-p	1.0	1.0
Max output tokens (Stage 1)	8192	8192
Max output tokens (Stage 2)	4096	4096
Request seed	0	0
Structured-output mode	JSON schema	JSON schema
Max retries / failed call	3	3
Stage-2 batch size	25	25

Material-identity atoms on UT-Zappos. The default applies to MIT-States and C-GQA. For UT-Zappos, the mandatory material-identity atom is exempt from the ambiguity/intersection filters; auxiliary appearance atoms remain subject to both. This preserves the label and gives full primitive coverage without asserting whole-object material.

The rule favors precision over coverage. If no atom survives, $\Omega(a) = \varnothing$; the example still trains the composition model but contributes no state loss, and candidates with $m(a) = 0$ receive no refinement.

C.5 Prompts and Inference Configuration

Both models receive identical prompts and schemas. Stage 1 constructs the smallest valid slot–value inventory; Stage 2 maps primitives to minimal, directly expressed atoms from that frozen inventory,

Table 12: Recovery execution summary. Stage 2 uses alphabetical batches of at most 25 primitives. All batches used downstream passed schema and deterministic validation.

Dataset	Recovery model	Stage-1 calls	Stage-2 calls	Retry calls	Validated batches
MIT-States	GPT-5.5	1	5	0	5
MIT-States	Gemini 2.5 Pro	1	6	1	5
UT-Zappos	GPT-5.5	1	1	0	1
UT-Zappos	Gemini 2.5 Pro	1	2	1	1
C-GQA	GPT-5.5	1	18	1	17
C-GQA	Gemini 2.5 Pro	1	19	2	17

Table 13: JSON schema contract and deterministic checks. The table records the fields and constraints relevant to validation without reproducing the schema listings.

Stage	Required fields	Field constraints	Cross-field validation
Stage 1	`slots`; each slot has `slot`, `definition`, `values`, and `exclusivity_note`.	Slot names and values use lowercase snake case. Definitions and notes are nonempty and at most 160 characters; values are nonempty and unique.	No duplicate slot names; every slot has at least one value; values are checked for mutual exclusivity and membership in the returned inventory.
Stage 2	`mappings`; each mapping has primitive, atoms, ambiguous, and `out_of_scope`. Each atom has `slot`, `value`, and `justification`.	At most three atoms per primitive; justifications are at most 140 characters. `competing_readings` is used when ambiguity is flagged.	Each batch contains every primitive exactly once; slots and values must be in the frozen inventory; no duplicate atom or two values from one slot; ambiguous mappings require at least two readings; both exclusion flags cannot be true.

Table 14: Primitive-level attrition under conservative consensus. "Empty intersection" includes valid mappings for which no atom survives agreement.

Dataset	Input primitives	Ambiguous	Out of scope	Empty intersection	Covered	Retained atoms
MIT-States	115	7	3	5	100	108
UT-Zappos	16	0	0	0	16	39
C-GQA	413	52	34	44	283	286

allowing an empty map. Table 10 lists the exact supplied fields and exclusions. No image, split, frequency, prediction, or result is provided.

Table 11 gives the inference settings. Both models use greedy decoding (temperature 0, top-p 1.0); schema failures are retried at most three times.

Recovery execution summary. Table 12 separates all Stage-2 calls, including retries, from the validated batches used for consensus.

C.6 Output Schema and Deterministic Validation

Table 13 summarizes the strict JSON fields, length limits, and cross-field checks for both stages.

Each Stage-2 batch must contain every requested primitive exactly once, and all batches must cover the vocabulary. Slots and values must belong to the frozen inventory, with no repeated atom or two values in one slot; ambiguous items require two competing readings, and the exclusion flags cannot co-occur. Aliases use the archived dictionary. Invalid outputs are retried under the same prompt; Tables 11 and 12 give the budget and observed counts. The validator does not resolve semantic

Table 15: Consensus inventories and mapping sizes. $|\mathcal{A}|$ and $|\mathcal{S}|$ count state primitives and slots; "Values" counts slot–value inventory entries. Mapping sizes $m = |\Omega(a)|$, retained assignments ("Atoms"), and the mean $\bar{m}$ over covered primitives are computed from the released observation-structure JSON.

Dataset	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{S}\rvert$	Values	$m=0$	$m=1$	$m=2$	$m=3$	Atoms	$\bar{m}$
MIT-States	115	26	102	15	92	8	0	108	1.08
UT-Zappos	16	5	26	0	1	7	8	39	2.44
C-GQA	413	27	283	130	280	3	0	286	1.01

Table 16: Representative entries from the released consensus mappings. Empty mappings receive neither partial-state supervision nor state refinement.

Dataset	Primitive	Consensus mapping $\Omega(a)$	Case
MIT-States	*wet*	(`moisture`, `wet`)	Single atom
MIT-States	*molten*	(`thermal_state`, `molten`); (`viscosity`, `runny`)	Multi-atom
MIT-States	*dull*	$\varnothing$	Ambiguous/ uncovered
UT-Zappos	*Canvas*	(`material_identity`, `canvas`); (`material_appearance`, `fabric_like`); (`weave`, `woven`)	Multi-atom
UT-Zappos	*Synthetic*	(`material_identity`, `synthetic`)	Identity only
C-GQA	*burnt*	(`thermal`, `burnt`)	Single atom
C-GQA	*greasy*	(`cleanliness`, `greasy`); (`surface`, `glossy`)	Multi-atom

disagreement: Eq. (15) removes it. It also rejects justifications based on what objects *usually*, *often*, or *tend to* look like.

Consensus attrition. Table 14 applies mutually exclusive exclusions in this order: ambiguity, out of scope, then empty intersection. UT-Zappos identity atoms are added afterward.

C.7 Recovered-Structure Statistics and Examples

Table 15 combines the inventory counts and mapping-size distribution; $\bar{m}$ averages retained atoms over covered primitives only. Uncovered primitives create neither an "other" value nor negative labels.

Most covered MIT-States and C-GQA primitives map to one atom. The eight MIT-States primitives with multiple retained atoms are *molten*, *mashed*, *pureed*, *muddy*, *wilted*, *frayed*, *eroded*, and *shattered*. Each mapping contains only agreed lexical facts. UT-Zappos is denser because material identity can coexist with surface or reflectance cues.

Table 16 shows representative single-atom, multi-atom, and uncovered mappings from all three datasets.

C.8 Human Verification Protocol

The audit scores retained atoms, not whole mappings. Atom (a, s, v) is correct only if s is appropriate for a, v is correct, and the assignment is lexically entailed rather than inferred from association or co-occurrence. Precision excludes empty mappings.

Aggregate audit summary. We sample retained atoms uniformly without replacement from five sources: human structure, each model, their consensus, and a random control. Duplicate atoms share one hidden-source judgment. The seed-2027 control assigns one to three valid inventory atoms to each of 111 primitives, matching the mapping-size range without using primitive–atom associations. Five annotators work independently; a senior annotator resolves non-unanimous cases.

Table 17: Atom-level human audit on MIT-States. An atom is correct only when its slot and value are directly entailed by the primitive. Precision uses adjudicated labels; agreement is measured before adjudication.

Structure	Retained atoms	Audited atoms	Correct	Prec. (%)	Pairwise agree. (%)	Adjudicated
Human	114	86	83	96.5	93.0	8
GPT-5.5	123	70	67	95.7	91.7	9
Gemini 2.5 Pro	119	85	80	94.1	89.9	12
Consensus	108	77	75	97.4	95.1	5
Random control	121	78	15	19.2	66.0	47

Table 18: POSS (Troika) implementation settings. "Effective batch" accounts for gradient accumulation. Only the learning rate and batch configuration differ across datasets. Other POSS instantiations retain their base-scorer optimization settings, while λ, γ, and the empirical reference definition remain shared POSS settings.

Setting	MIT-States	UT-Zappos	C-GQA
Backbone	ViT-L/14	ViT-L/14	ViT-L/14
Context length	8	8	8
Adapter bottleneck	64	64	64
Adapter dropout	0.1	0.1	0.1
Optimizer	AdamW	AdamW	AdamW
Learning rate	1.0×10^{-4}	5.0×10^{-4}	3.125×10^{-6}
Weight decay	1.0×10^{-5}	1.0×10^{-5}	1.0×10^{-5}
Physical / effective batch	8 / 16	16 / 32	2 / 4
Maximum epochs	30		
LR schedule	StepLR, step size 5, multiplier 0.5		
Model selection	early stopping on validation AUC		
λ / γ	1.0 / 0.5		
Reference distribution	empirical training reference		

Table 19: Computational overhead under a shared environment. Measurements use MIT-States (30,338 training images) and a single RTX 3090 isolated from other jobs. The base scorer uses a physical batch size of 16. Base scorer + POSS uses a physical batch size of 8 and accumulates gradients over two steps to keep the same effective batch size, because the state branch's per-object broadcast tensor otherwise exceeds GPU memory.

Method	Train time	Images/s
Base scorer	84.1 min/epoch	6.02
Base scorer + POSS	87.9 min/epoch	5.75

Table 17 shows that consensus has the highest precision despite lower coverage, while the random control is substantially worse. The audit evaluates fixed structures only; it does not tune mappings, hyperparameters, or POSS.

D Experimental and Evaluation Details

D.1 Common Training Setup

All experiments use the standard MIT-States (Isola et al., 2015), UT-Zappos (Yu & Grauman, 2014), and C-GQA (Naeem et al., 2021) splits with frozen CLIP ViT-L/14 (Radford et al., 2021). Only the visual adapter (Chen et al., 2022), soft prompts, and POSS modules are trainable. The consensus structures are fixed across architectures, hyperparameters, and seeds; only the explicit structure and coverage controls alter them. Training runs use PyTorch 2.3 on Ubuntu 20.04 on a workstation

Table 20: Composition of the MIT-States multi-state diagnostic subset.

Subset statistic	Value
Images screened	900
Images retained	612
Original-slot labels	612
Additional-slot labels	938
Distinct slots represented	22
Distinct slot–value atoms represented	71
Mean additional labels per retained image	1.53
Annotators per image	3
Mean pairwise agreement	88.6%

equipped with two NVIDIA RTX A6000 GPUs. The isolated RTX 3090 environment used only for the computational-overhead measurements is reported separately in Table 19.

Table 18 gives the POSS (Troika) settings; effective batch includes accumulation. AdamW and StepLR run for at most 30 epochs, with early stopping on validation AUC. Learning rate and batch size follow each base Troika configuration; λ and γ are shared across datasets.

POSS keeps the base candidate space and composition loss: the base supplies $F_{\psi,\phi}$ and POSS adds $G_{\theta,\phi}$. Troika (Huang et al., 2024) and DFSP (Lu et al., 2023) retain their base settings unless listed otherwise; the plain POSS row uses the CoOp-style (Zhou et al., 2022) scorer from the main paper.

D.2 Empirical Reference Distribution

The reference $\bar{q}_s$ uses only training labels $a_1, \ldots, a_N$ and the frozen observation map. For $v \in \mathcal{V}_s$, define

$$n_s(v) = \sum_{i=1}^{N} \mathbb{I}[(s, v) \in \Omega(a_i)]. \tag{16}$$

Each retained atom receives one count per image carrying its primitive. With add-one smoothing,

$$\bar{q}_s(v) = \frac{n_s(v) + 1}{\sum_{v' \in \mathcal{V}_s} n_s(v') + |\mathcal{V}_s|}. \tag{17}$$

The distribution is frozen before optimization and enters the candidate score. It does not change the partial-state likelihood; through the fused training logits it can also affect composition-branch optimization.

D.3 Random Seeds and Uncertainty Reporting

All POSS variants use seeds $\{0, 1, 2\}$. Metrics are evaluated separately for each run and summarized as means and sample standard deviations (denominator $n - 1$) in percentage points.

Transfer intervals use 10,000 seed-2027 bootstrap samples of unseen compositions. Within each resample, gains are weighted by test-image count. The interference study instead resamples complete update sequences, preserving all time points (10,000 samples; seed 2027).

D.4 Multi-State Diagnostic Annotation

The multi-state diagnostic tests visible states outside the slot named by the standard MIT-States label. It uses held-out images only; diagnostic labels do not affect training, selection, recovery, or calibration.

Diagnostic cohort. We screen nine images for each of 100 covered primitives (900 total). Three annotators work independently, and a senior annotator resolves cases without a majority. Table 20 reports the adjudicated cohort and pre-adjudication pairwise agreement.

Table 21: Three-seed results for all POSS instances. Entries report mean±sample standard deviation in percentage points. CW and OW denote closed- and open-world evaluation.

POSS instance	Dataset	Protocol	AUC	HM	S	U
POSS	MIT-States	CW	24.6 ± 0.18	41.7 ± 0.29	51.6 ± 0.63	56.1 ± 0.82
POSS	MIT-States	OW	9.6 ± 0.05	23.1 ± 0.16	51.5 ± 0.91	23.7 ± 0.42
POSS	UT-Zappos	CW	46.3 ± 0.38	58.5 ± 0.61	70.5 ± 2.31	78.9 ± 2.21
POSS	UT-Zappos	OW	37.9 ± 0.54	52.7 ± 0.52	70.3 ± 2.42	66.5 ± 1.32
POSS	C-GQA	CW	19.2 ± 0.27	37.3 ± 0.39	45.8 ± 1.08	44.8 ± 1.01
POSS	C-GQA	OW	5.7 ± 0.12	17.8 ± 0.24	45.7 ± 1.18	13.5 ± 1.21
POSS (DFSP)	MIT-States	CW	25.0 ± 0.20	42.1 ± 0.33	52.2 ± 0.58	56.0 ± 0.85
POSS (DFSP)	MIT-States	OW	9.9 ± 0.06	24.0 ± 0.18	51.8 ± 0.82	23.8 ± 0.45
POSS (DFSP)	UT-Zappos	CW	47.0 ± 0.42	59.4 ± 0.65	72.6 ± 2.18	79.4 ± 2.05
POSS (DFSP)	UT-Zappos	OW	38.9 ± 0.56	53.8 ± 0.56	71.8 ± 2.33	67.3 ± 1.27
POSS (DFSP)	C-GQA	CW	19.4 ± 0.29	37.5 ± 0.34	46.4 ± 1.01	43.6 ± 0.93
POSS (DFSP)	C-GQA	OW	5.9 ± 0.13	18.2 ± 0.21	46.2 ± 1.15	12.9 ± 1.17
POSS (Troika)	MIT-States	CW	25.8 ± 0.19	42.8 ± 0.31	53.6 ± 0.80	55.0 ± 1.09
POSS (Troika)	MIT-States	OW	10.3 ± 0.04	24.2 ± 0.17	53.4 ± 1.02	23.0 ± 0.37
POSS (Troika)	UT-Zappos	CW	50.4 ± 0.44	61.9 ± 0.72	74.7 ± 2.56	80.7 ± 2.64
POSS (Troika)	UT-Zappos	OW	40.8 ± 0.60	55.5 ± 0.58	72.7 ± 2.64	68.2 ± 1.40
POSS (Troika)	C-GQA	CW	19.9 ± 0.31	38.2 ± 0.42	46.6 ± 1.22	46.3 ± 1.13
POSS (Troika)	C-GQA	OW	6.1 ± 0.14	18.8 ± 0.27	46.4 ± 1.37	14.1 ± 1.35

Table 22: Seen/unseen accuracies for the reference-distribution comparison in Table 5(b). Entries are MIT-States means over three seeds.

Evidence reference	Closed world		Open world	
	S	U	S	U
Uncentered	48.1	41.0	46.9	12.4
Uniform	50.9	54.1	50.8	21.7
Empirical (default)	51.6	56.1	51.5	23.7
Max centering	51.3	55.2	51.2	22.6
Uncentered + learned offset	49.3	50.6	49.0	18.7

Annotators see the image, object, inventory, and slot definitions, but no prediction. They verify the original state and mark clearly visible values in other slots; occluded or uncertain slots remain unlabeled, not negative. The subset requires a verified original state and at least one verified value in another slot.

Prediction is restricted to each labeled slot. Observed accuracy uses the benchmark slot; unobserved accuracy and NLL use verified additional slots. All are computed per slot and macro-averaged, so frequent properties do not dominate. These metrics evaluate state evidence, not composition accuracy.

E Additional Quantitative Results

E.1 POSS Variants Across Three Random Seeds

Three-seed results. Table 21 expands every POSS row in the main comparison and reproduces each mean after one-decimal rounding. It reports all four metrics by variant and protocol.

E.2 Reference-Distribution Ablation

Table 22 supplies the best seen and unseen accuracies for the reference choices in Table 5(b), under the same three-seed protocol.

Table 23: Seen/unseen accuracies for the coverage experiment in Table 6. Entries are MIT-States means over three seeds.

Drop	Remaining atoms	Closed world		Open world	
		S	U	S	U
0%	108	51.6	56.1	51.5	23.7
25%	81	50.9	54.6	50.9	22.3
50%	54	49.8	52.1	49.7	20.1
75%	27	48.5	49.2	48.6	17.5
100%	0	46.5	45.0	46.0	10.0

Table 24: Aggregate reproducibility record for the MIT-States gradient audit. Each variant processes 64 batches of 16 images; valid and zero-denominator counts cover all 1,024 samples. The final column reports off-slot magnitude and conflict percentages using the row-specific gradient source.

Variant	Gradient source	Batches	Valid	Zero denom.	Off-slot / conflict
Dense-negative	Dense state loss	64	1008	16	42.0 / 55.0
No-SG	Composition path	64	1000	24	43.5 / 45.0
POSS	Partial-state loss	64	1024	0	0.0 / –

E.3 Coverage Stress Test

For the coverage experiment in Table 6, masks use seed 3407 and are reused across training seeds $\{0, 1, 2\}$. Table 23 adds the best seen and unseen accuracies. The 0% row is full POSS; the 100% row removes all state supervision and evidence scoring while retaining the composition scorer. This control tests reduced coverage rather than incorrect mappings.

F Diagnostic Protocols and Numerical Tables

F.1 Numerical Cross-Slot Gradient Audit

For each row, let ℓ_{src} denote the row-specific audit source and let $g_t = \nabla_{\theta_t} \ell_{\mathrm{src}}$. Dense-negative uses its dense state loss, POSS uses the partial-state loss, and No-SG uses the composition loss through the state-evidence path. The numerator and denominator are always computed from the same source. For revealed slots $S(a)$, define

$$r_{\mathrm{off}} = \frac{\sum_{t \notin S(a)} \|g_t\|_1}{\sum_{t \in S(a)} \|g_t\|_1}. \tag{18}$$

Zero denominators are counted and excluded. Conflict means negative cosine similarity with the first same-batch supervised gradient for that slot; updates without such a reference are excluded. We audit 64 fixed MIT-States mini-batches of 16 images. All 1,024 audit samples are covered examples with $\Omega(a) \neq \varnothing$. Shared-adapter gradients are outside this measurement.

For POSS, the partial-state loss gives zero gradient to slots outside $S(a)$ while retaining supervised gradients on revealed slots. Its off-slot ratio is therefore zero with a valid denominator. Separately, stop-gradient makes the direct composition-to-state gradient identically zero; this direct path is not used to normalize Eq. (18).

Table 24 reports the batch, valid-sample, zero-denominator, off-slot, and conflict statistics for all three variants.

Gradient-audit results. Dense-negative and No-SG have off-slot ratios of 42.0% and 43.5%; 55.0% and 45.0% of valid updates conflict. POSS has a zero off-slot ratio under the partial-state loss. Its direct composition-to-state path is also zero by construction, so no conflict rate is defined.

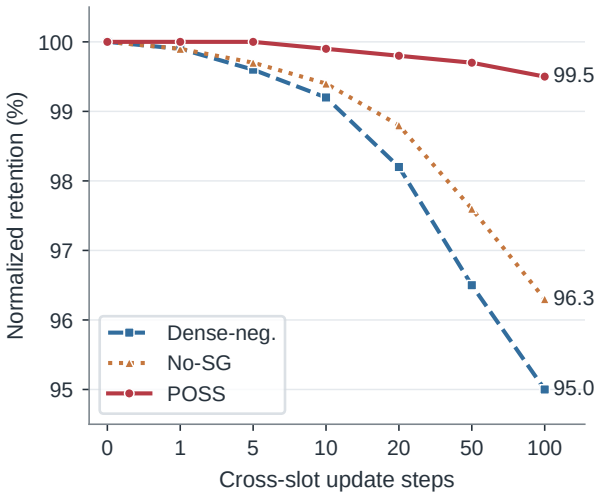


Figure 5: Controlled retention of target-slot predictions during updates on another slot of the same object. All variants start from a common checkpoint; accuracy is normalized to the value before the updates. Ticks enumerate measured steps at equal spacing. Full settings and numerical values are given in the accompanying tables.

Table 25: Fixed settings for controlled cross-slot interference.

Controlled-update setting	Value
Selected sequences	10
Conditions per sequence	3
Minimum object–slot support	8 images
Update batch size	4
Update learning rate	1.0×10^{-4}
Update steps	100
Evaluation steps	0, 1, 5, 10, 20, 50, 100
Bootstrap samples / seed	10,000 / 2027

Table 26: Target-slot retention over controlled update steps on MIT-States. Entries are normalized accuracies and reproduce the point values plotted in Figure 5.

Variant	Update condition	0	1	5	10	20	50	100
Dense-negative	Same-object cross-slot	100.0	99.9	99.6	99.2	98.2	96.5	95.0
No-SG	Same-object cross-slot	100.0	99.9	99.7	99.4	98.8	97.6	96.3
POSS	Same-object cross-slot	100.0	100.0	100.0	99.9	99.8	99.7	99.5
POSS	Same-slot control	100.0	100.0	99.9	99.9	99.8	99.7	99.6
POSS	Different-object control	100.0	100.0	100.0	100.0	99.9	99.9	99.8

F.2 Controlled Cross-Slot Interference

All variants start from the same trained checkpoint. We freeze the composition scorer and update only the state branch for 100 steps. Closed-world top-1 is measured on held-out images of a target slot. Updates use the target slot, another slot of the same object, or that slot on another object; the latter two isolate same-object cross-slot interference from generic drift.

Variants and conditions share data count, optimizer, and steps, and restart from that checkpoint. Accuracy at steps $0, 1, 5, 10, 20, 50, 100$ is normalized by step 0. The first ten sorted eligible sequences are fixed across variants; each pair has at least eight training images and an available different-object control.

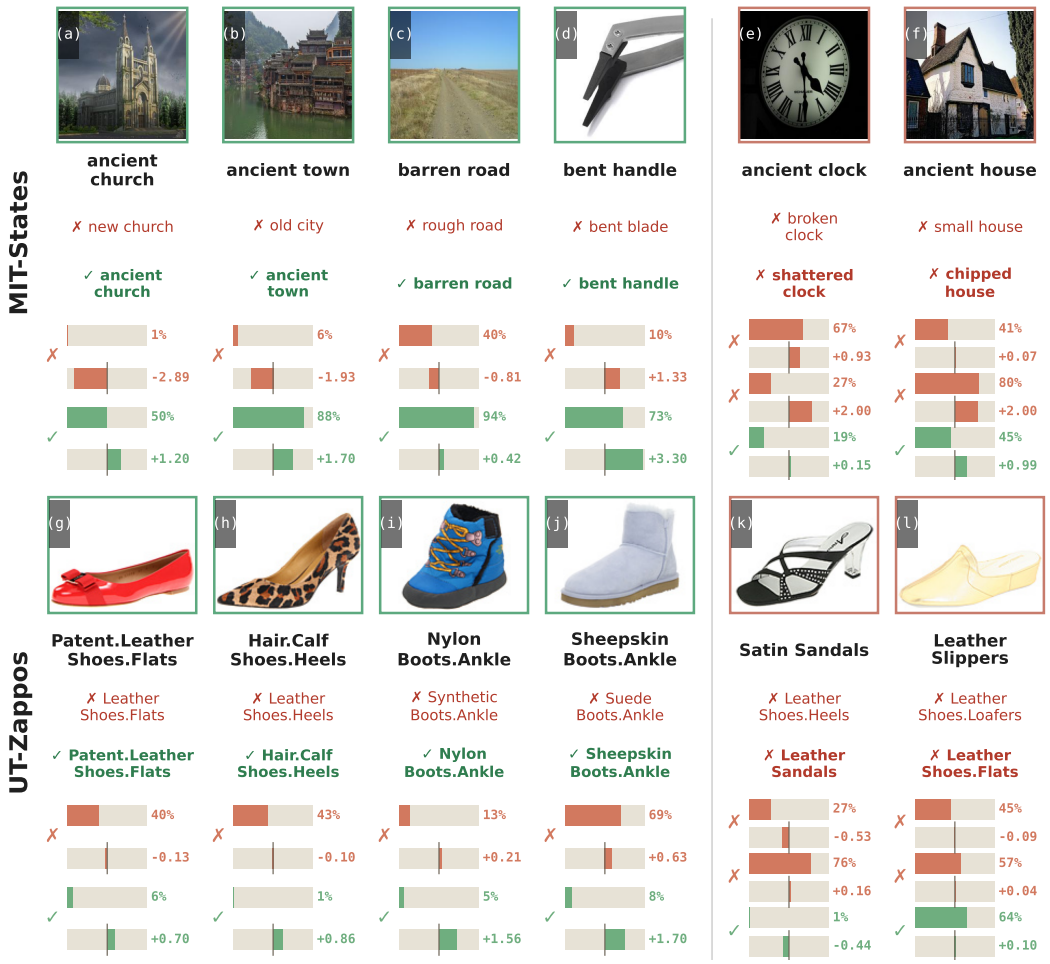


Figure 6: Qualitative effects of state-evidence fusion on real test images. Each case lists the ground truth, composition-only prediction, and fused prediction, in that order. Green borders mark corrections; red borders mark changed but still incorrect predictions. Check and cross marks identify ground-truth and incorrect candidate states. The upper bar shows the raw posterior of the first mapped atom (0–100%); the lower bar shows the reference-corrected evidence $G_{\theta,\phi}$, averaged over all mapped atoms. Evidence bars use a common $[-3.5, 3.5]$ scale centered at zero; positive values favor the candidate relative to its training frequency.

Table 25 records the controlled-update settings, and Table 26 gives the normalized retention values underlying Figure 5.

Controlled-interference results. After 100 same-object cross-slot updates, retention is 95.0% for dense-negative, 96.3% for No-SG, and 99.5% for POSS. POSS retains 99.6% and 99.8% in the same-slot and different-object controls. These normalized values measure retention, not absolute accuracy.

F.3 Qualitative Case Analysis

We compare closed-world predictions from the seed-0 MIT-States and UT-Zappos checkpoints using the composition score alone, $F_{\psi,\phi}$, and the fused score, $F_{\psi,\phi} + \gamma G_{\theta,\phi}$. Both scores come from the same POSS checkpoint, isolating the ranking change produced by adding state evidence.

A correction candidate has an incorrect composition-only prediction and a correct fused prediction. A residual-failure candidate also starts from an incorrect prediction and is changed by fusion, but

remains incorrect. Full-test-set inference yields 190 corrections and 1,790 residual failures on MIT-States, and 62 corrections and 209 residual failures on UT-Zappos. Figure 6 shows four corrections and two residual failures per dataset, selected while preferring distinct ground-truth compositions. These examples illustrate mechanisms rather than estimate their prevalence. These counts omit correct-to-incorrect changes and therefore do not quantify the net accuracy effect of fusion.

In the corrected cases, the ground-truth attribute generally receives stronger reference-corrected evidence than the composition-only alternative. Case (g) shows why centering matters: the raw posterior is 6% for *Patent.Leather* and 40% for the common but incorrect *Leather*, whereas their corrected evidence is $+0.70$ and -0.13, respectively. The residual failures expose two different limits. In case (e), the state branch actively favors incorrect attributes over *ancient*; in case (k), all relevant evidence is weak or near its reference level. Thus fusion can repair composition errors when state evidence is discriminative, but it can still mislead or provide too little signal when the state posterior is wrong or uncertain.

G Limitations and Scope

The main limitations are incomplete observation and limited scope. Consensus can inherit lexical bias and leave primitives uncovered, while recovering the full slot posterior additionally requires conditional ignorability of slot revelation; otherwise the model targets the revealed subpopulation. The slot-wise factorization omits arbitrary cross-slot dependence, and stop-gradient blocks the direct composition-to-state path while the shared visual adapter keeps the objectives coupled. State-prediction errors enter composition decisions through the evidence score, as illustrated by the residual errors in Appendix F.3. Rare atoms and a weak base scorer can limit transfer. On MIT-States, its measured overhead is 4.5% more epoch time and 4.5% lower throughput under the fixed effective batch; Table 19 reports the measurement details.

H Reproducibility Statement

Section 4 specifies the predictors, objectives, and inference rule. Appendix C provides the recovery inputs, prompts, schemas, and validation procedure; the accompanying `Supp/` files contain the recovered mappings. Training settings, reference frequencies, random seeds, and diagnostic protocols are described in Appendices D and F. Appendix A states the assumptions behind the likelihood and posterior interpretations. Figure 4 reports hyperparameter sensitivity; the additional tables provide coverage, uncertainty, and computational-cost measurements.