AFTERIMAGE: A DISCRETE ADAPTER FOR INSPECTABLE MULTIMODAL FUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretrained multimodal large language models (MLLMs) are increasingly used to *fuse* heterogeneous inputs into a dense, informative latent representation consumed by a downstream model. However, which domain features the fusion carries remains opaque, which limits transparency in high-stakes deployments and leaves unclear whether a prediction reflects the downstream decoder or the quality of the fusion. Prior interpretation approaches usually explain the model after training, such as through attention maps or probes, rather than recording which domain features the fusion itself carries. We present AFTERIMAGE, an adapter that makes part of a frozen MLLM's fusion inspectable on the forward path. AFTERIMAGE discretizes the residual stream at several depths into short code sequences from one shared codebook, giving a discrete inspectable record at each depth. From these records, we extract code motifs and verify them as associations with domain features. With only 0.20–0.71% added parameters, AFTERIMAGE both lets the domain features its own write carries be read from its codes and matches or improves task performance. AFTERIMAGE lowers counterfactual-age FID on 3D brain MRI synthesis from 1.93 to 0.43, raises $\pi_{0.5}$ robot success by 26.0 points averaged over shifted LIBERO-Pro suites, and improves video question answering accuracy by 16.0 points over the frozen backbone. On withheld samples, the extracted motifs expose associations with features such as clinical diagnosis, demographic information, objects in the scene, and robot task identity. Substitution tests then support a causal role for these motifs, as overwriting them changes the corresponding features in the output, and swapping two robot task motifs redirects the policy toward the other task. Recorded during fusion, the motifs give a per-input account of which domain features the adapter's write carried, and that account is associated with how the model then behaves.

1 INTRODUCTION

Pretrained multimodal large language models (MLLMs) are increasingly used not only to generate text but to fuse heterogeneous inputs into one latent representation consumed by a downstream decoder, policy, or answer head (Tsimpoukelli et al., 2021). Language models are pretrained by next-token prediction on large corpora, and multimodal models extend this knowledge by aligning further modalities to the same representation (Radford et al., 2021; Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023b). The downstream model reads the backbone's hidden states as the fused representation to produce the task output. For instance, an action expert predicts robot actions by attending to the hidden states of a pretrained vision-language model given a text instruction and camera views (Brohan et al., 2023; Beyer et al., 2024; Shukor et al., 2025; Physical Intelligence, 2025). The backbone stays frozen or lightly adapted, as retraining is costly, and a small trained component, such as an adapter, a low-rank update, or a learned-query resampler, connects the backbone to the downstream model (Houlsby et al., 2019; Hu et al., 2021; Li et al., 2023; Jaegle et al., 2021).

Fusion through a pretrained backbone underlies widely used systems for vision-language understanding (Tsimpoukelli et al., 2021; Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023b; Bai et al., 2025), conditional image and medical image generation (Saharia et al., 2022; Rombach et al., 2022; Pinaya et al., 2022), video understanding (Gemini Team et al., 2024; Bai et al., 2025), and embodied planning and robot control (Driess et al., 2023; Brohan et al., 2023; Beyer et al., 2024; Shukor et al., 2025; Physical Intelligence, 2025). However, the fused representation is a dense hidden state whose

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.

domain features involved remain implicit. Therefore, the quality of the final prediction is hard to attribute to the strength of the downstream model or to error accumulated in the fusion (Alain & Bengio, 2016; Belinkov, 2022). The ambiguity matters most where errors are costly. For example, when a brain MRI generator given an age of 80 produces a scan without the ventricular enlargement usually expected in old age, the trained component connecting the backbone to the generator may have dropped the age, or the generator may have ignored the age. A readable record of what the trained component adds to the fusion would show whether the age survived in that contribution, and editing the record would test whether the generator acts on the contribution.

Two families of methods recover the domain features carried by the fused representation. First, natural-language explanations elicited from the model are expressed in readable terms, whereas the explanations are produced after the fusion and require a language-modeling head unused by a backbone serving a diffusion decoder or a policy (Radford et al., 2021; Rombach et al., 2022; Shukor et al., 2025). Controlled studies also find that explanations can omit the features determining the output (Turpin et al., 2023; Lanham et al., 2023). Second, post-hoc analyses, such as attention maps, linear probes, and sparse autoencoders, read the backbone's hidden states directly (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019; Alain & Bengio, 2016; Belinkov, 2022; Cunningham et al., 2023; Bricken et al., 2023), whereas the recovered structure is fitted after training and expressed in the model's own units rather than the domain's. Both families therefore describe the fusion from outside the model's computation, and a record produced within the computation itself remains open.

We propose AFTERIMAGE, an adapter inserted inside the frozen MLLM, which exposes the domain features carried by the adapter's contribution in three steps (Figure 1). First, the adapter's contribution to the fusion is recorded as a short, ordered sequence of discrete codes. At each of several depths, one adapter with shared weights gathers the residual stream into a small number of ordered slots, quantizes each slot to an entry of a learned codebook, and writes the decoded codes back into the stream, leaving every pretrained layer unchanged. The write to the residual stream depends on the input only through the selected codes. For fixed parameters and input length, the code sequence therefore determines the adapter's contribution, and the record lies on the deployed forward path. Second, after training, a motif set is built from short consecutive code runs selected for association with each domain feature and validated on held-out inputs. Sharing one codebook across depths and learning the codebook with the downstream task fixes discrete units during training and makes short motifs comparable across inputs and depths. Third, at inference, the record of a new input is compared against the motif set, which indicates the domain features associated with the write. We evaluate AFTERIMAGE on brain MRI synthesis, robot action planning, and video question answering. Across the three domains, AFTERIMAGE matches or improves performance compared to baselines with under 0.71% added parameters. In addition, the motifs selected after training are associated with diagnosis, age, and robot task identity on held-out inputs. We further test the associations by intervention through motif substitution. Experiment results show that, overwriting one motif with the motif of another feature value, while the rest of the model stays unchanged, moves the output of the downstream model toward the substituted value.

Our contributions are as follows.

- We introduce AFTERIMAGE, a discrete adapter inserted inside a frozen MLLM backbone, which records the adapter's contribution to the fusion as discrete codes associated with domain features.
- With 0.20–0.71% added parameters, AFTERIMAGE matches or improves on the evaluated baselines in all three domains, and the margins widen on the tasks requiring generalization.
- Our experiments show that AFTERIMAGE code motifs are associated with domain features, such as diagnosis and robot task, on held-out inputs. Motif substitution tests further show that the motifs influence the corresponding features in the downstream model's output.

2 AFTERIMAGE

AFTERIMAGE, illustrated in Figure 1, is a discrete adapter designed to expose the domain features involved in the fusion inside a frozen MLLM backbone. The adapter records its contribution to the fusion as a short sequence of discrete codes on the forward path, leaving every pretrained weight fixed. After training, short runs of consecutive codes, called motifs (e.g., trigrams), are extracted and

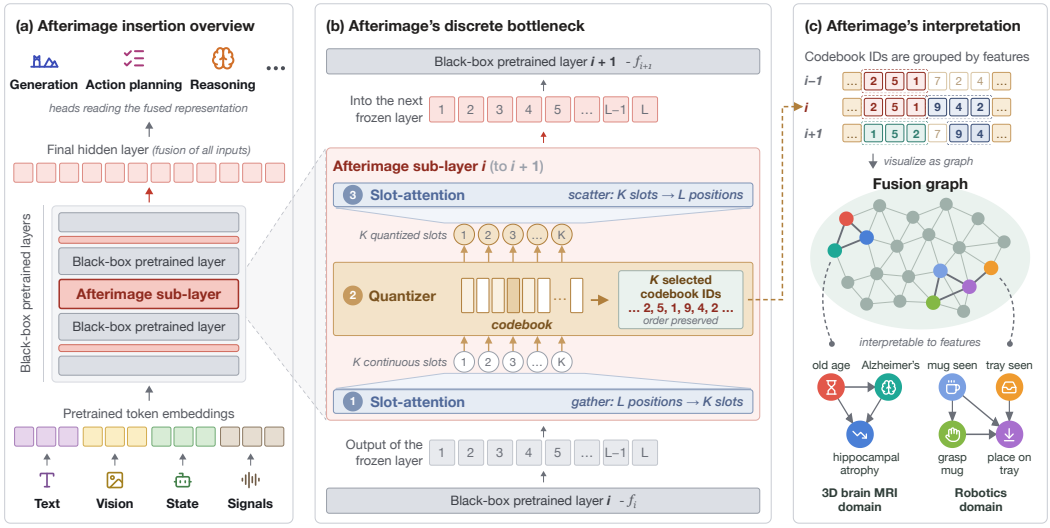


Figure 1: **Overview of AFTERIMAGE. (a)** One shared discrete adapter is inserted at several depths of a frozen backbone. **(b)** At each depth the adapter gathers the stream into K ordered slots, quantizes each slot against a shared codebook, and scatters the codes into a write. **(c)** Code motifs are read across depths and tested for association with domain features.

tested for association with domain features. The motifs are then used to read the domain features of held-out inputs and to edit an individual input, testing whether the downstream model responds.

2.1 PROBLEM SETUP

Given raw inputs $\mathcal{X} = \{x_1, \ldots, x_N\}$ from heterogeneous sources, a pretrained encoder ψ_n maps each x_n to L_n tokens of width d, and the tokens are concatenated into $h^{(0)} \in \mathbb{R}^{L \times d}$ with $L = \sum_n L_n$. A frozen backbone $F = f_B \circ \cdots \circ f_1$ fuses the tokens along a residual stream, $h^{(i)} = f_i(h^{(i-1)})$ (Radford et al., 2021; Li et al., 2023). A task-specific model $\mathcal{T}$ (a generator, policy, or answer head, trained or frozen per domain) reads the output of F through Pool, a cross-attention with N_q learned queries or the backbone's own output head, under $\mathcal{L}_{\text{task}}$ (Equation 1, Appendix C).

$$\hat{y} = \mathcal{T}\big(\text{Pool}(F(h^{(0)}))\big), \qquad \mathcal{L}_{\text{task}} = \mathbb{E}_{(\mathcal{X}, y^*) \sim \mathcal{D}}\big[\ell_{\text{task}}(\hat{y}, y^*)\big]. \tag{1}$$

2.2 DESIGN PRINCIPLES

Four properties guide the design, and Appendix B compares prior designs against them. **P1 (on-path).** The record is produced in the deployed model's forward pass, as a description fitted afterwards can diverge from the computation, and the adapter therefore sits in the residual stream. **P2 (recoverable write).** An edit to the record is a controlled edit to the write only if nothing the adapter contributes bypasses the codes. The write is therefore decoded from the codes alone. **P3 (comparable across depths).** Following a motif across depths requires a code ID to denote the same entry everywhere, and one codebook and one set of weights serve each depth. **P4 (ordered).** A fused input is a composition, which a sequence expresses and a set does not. The slots are therefore refined left to right and decoded with position embeddings.

2.3 THE AFTERIMAGE ADAPTER

At each insertion depth $i \in \mathcal{S} \subset \{1, \ldots, B\}$, the adapter $\mathcal{A}$ outputs K codebook IDs $\kappa^{(i)} = (k_1, \ldots, k_K)$ and a write $w^{(i)} \in \mathbb{R}^{L \times d}$ read by the next frozen layer. The adapter is composed of three stages, $\mathcal{A} = \text{Scatter} \circ \text{Quantize} \circ \text{Gather}$, which map the projected stream $u^{(i)} = h^{(i)} W_{\downarrow} \in \mathbb{R}^{L \times d_a}$ through $K \ll L$ slots. Gather compresses the stream into K ordered slots, Quantize assigns each slot to a codebook entry, and Scatter decodes the codes into the write. The same module, the same projections $W_{\downarrow} \in \mathbb{R}^{d \times d_a}$ and $W_{\uparrow} \in \mathbb{R}^{d_a \times d}$, and the same codebook $E \in \mathbb{R}^{V \times d_a}$ with rows e_v serve each depth (P3), and adapters at several depths give a record at each depth (Appendix D.2). The pretrained backbone remains frozen throughout training, and only the adapter $\mathcal{A}$, the projections

$W_\downarrow$ and $W_\uparrow$, and the codebook E are learned. The pooling module Pool is either trained jointly or inherited frozen from an earlier training stage, depending on the domain (Appendix D.3). For brevity, the depth superscript is omitted in the following.

Gather. Gather compresses the stream into K ordered slots, where K learned queries $Q \in \mathbb{R}^{K \times d_a}$ cross-attend to u, as in a latent-query resampler (Jaegle et al., 2021; Li et al., 2023), and a causally masked self-attention over the slot index refines the result, where MHA is cross-attention and the causal mask lets slot j attend to slots $1, \ldots, j$ (P4).

$$Z = \mathrm{MSA}_{\mathrm{causal}}\big(\mathrm{MHA}(Q,\, u)\big) \in \mathbb{R}^{K \times d_a}. \tag{2}$$

Quantize. Quantize assigns each slot its nearest codebook entry under cosine similarity and passes the K IDs onward with gradients through the straight-through estimator (Bengio et al., 2013), where a hat denotes an ℓ_2-normalized vector, a tilde a quantized one, and sg the stop-gradient. Assignment is deterministic, and the codes match between training and inference (Appendix D.1).

$$k_j = \arg\max_v \hat{z}_j^\top \hat{e}_v, \qquad \tilde{z}_j = \hat{z}_j + \mathrm{sg}\big[\hat{e}_{k_j} - \hat{z}_j\big], \tag{3}$$

Scatter and routing. Scatter decodes the quantized slots $\tilde{Z} \in \mathbb{R}^{K \times d_a}$ (rows $\tilde{z}_j$), tagged with slot-position embeddings $P_{\mathrm{slot}} \in \mathbb{R}^{K \times d_a}$, into the write through L learned position queries $U \in \mathbb{R}^{L \times d_a}$, and routing sets what the next layer reads,

$$a = \mathrm{FFN}\big(\mathrm{MHA}(U,\, \tilde{Z} + P_{\mathrm{slot}})\big) \in \mathbb{R}^{L \times d_a}, \tag{4}$$

$$w = a\, W_\uparrow, \qquad \tilde{h} = \alpha\, h + \beta\, w. \tag{5}$$

Inputs are padded to a common L per domain, as U is sized by L (Appendix C). Residual routing, $(\alpha, \beta) = (1, 1)$, adds the write to the intact stream and serves a downstream model conditioned on the full stream through Pool. Under residual routing, the record therefore accounts for the write alone. Exclusive routing, $(\alpha, \beta) = (0, 1)$, instead passes only the write to the next frozen layer and serves a downstream model attending to the keys and values of each layer, in which case the codes mediate what the downstream model reads at that depth.

Training objective. The task objective is optimized jointly with commitment and coverage terms summed over insertions. The commitment term keeps each slot near its entry, the coverage term counters codebook collapse, and the codebook is maintained by exponential moving averages (Appendix D.1, Algorithm 1).

$$\mathcal{L} = \mathcal{L}_{\mathrm{task}} + \textstyle\sum_{i \in \mathcal{S}} \big(\lambda_{\mathrm{commit}}\, \mathcal{L}^{(i)}_{\mathrm{commit}} + \lambda_{\mathrm{cov}}\, \mathcal{L}^{(i)}_{\mathrm{cov}}\big). \tag{6}$$

2.4 Reading and intervening on the codes

Both reading and substitution rest on one property of the adapter, that the write at an insertion is determined by the code sequence at that insertion. The property follows from the construction of Scatter and makes a code substitution a well-defined intervention on the write.

Proposition 1 (code-determined write)**.** *For fixed adapter parameters and length L, $w^{(i)} = g(\kappa^{(i)}, L)$ for a deterministic g, and under exclusive routing every layer from f_{i+1} up to the next insertion is likewise a function of $\kappa^{(i)}$ and L.*

Proof sketch. The straight-through estimator sets $\tilde{z}_j = \hat{e}_{k_j}$, making $\tilde{Z}$ a function of $\kappa^{(i)}$. Scatter reads only $\tilde{Z}$, the parameters P_{slot} and $W_\uparrow$, and U, a learned parameter sized by L. With $\alpha = 0$, $\tilde{h}^{(i)} = w^{(i)}$, and each frozen layer is a fixed function of its input. □

Reading proceeds after training through signatures that leave the model unchanged. A trigram $\tau = (i, k_j, k_{j+1}, k_{j+2})$ is a three-code run at depth i, $\mathrm{pres}(\tau, g)$ is the fraction of inputs in group g whose record contains τ, and the specificity of τ for g is $\mathrm{pres}(\tau, g) - \max_{g' \neq g} \mathrm{pres}(\tau, g')$. A signature is the trigram pair maximizing joint specificity on selection inputs, scored on disjoint inputs against a label-permutation null (Appendix H). A substitution overwrites $\kappa^{(i)}[j{:}j{+}3] \leftarrow \tau_{\mathrm{target}}$ wherever τ_{source} is realized, at each decision in sequential domains, and later depths are recomputed from the modified stream (Appendix H).

Table 1: **Brain MRI synthesis and counterfactual editing.** FID, age MAE, and sex accuracy for synthesis (Syn) and for CF edits of age (CF-A), sex (CF-S), and diagnosis (CF-C), grouped by the CN and Impaired cohorts of the conditions. The Real data row gives reference values on real scans. Trainable arms report mean±std over three runs, and bold marks the best mean. AFTERIMAGE achieves the best FID, age MAE, and sex accuracy across synthesis and CF in both cohorts.

	FID ↓						Age MAE (y) ↓					Sex accuracy (%) ↑				
	CN				Impaired		CN			Impaired		CN			Impaired	
Model	Syn	CF-A	CF-S	CF-C	Syn	CF-C	Syn	CF-A	CF-S	Syn	CF-C	Syn	CF-A	CF-S	Syn	CF-C
Real data	0.003				0.001		7.0			7.0		85.8			86.2	
🔥MLP + 🔥LDM	$1.38_{\pm.11}$	$1.91_{\pm.18}$	$1.46_{\pm.09}$	$1.35_{\pm.14}$	$1.24_{\pm.16}$	$1.11_{\pm.12}$	$19.4_{\pm1.4}$	$15.9_{\pm1.1}$	$15.4_{\pm0.9}$	$17.5_{\pm1.6}$	$16.5_{\pm1.2}$	$59.4_{\pm1.3}$	$58.6_{\pm0.8}$	$58.8_{\pm1.1}$	$64.5_{\pm1.5}$	$64.5_{\pm0.9}$
❄LLM + 🔥LDM	$1.21_{\pm.08}$	$1.93_{\pm.19}$	$1.61_{\pm.13}$	$1.43_{\pm.07}$	$0.80_{\pm.15}$	$1.35_{\pm.17}$	$14.0_{\pm0.8}$	$16.1_{\pm1.3}$	$15.6_{\pm1.0}$	$13.9_{\pm1.5}$	$14.1_{\pm1.1}$	$67.7_{\pm1.0}$	$65.6_{\pm1.4}$	$71.5_{\pm0.7}$	$79.8_{\pm1.2}$	$80.0_{\pm1.6}$
🔥Cont. + ❄LLM + ❄LDM	$1.16_{\pm.10}$	$1.47_{\pm.14}$	$1.03_{\pm.06}$	$1.32_{\pm.11}$	$1.48_{\pm.18}$	$1.43_{\pm.09}$	$12.7_{\pm0.9}$	$14.5_{\pm1.2}$	$12.3_{\pm1.0}$	$13.4_{\pm0.8}$	$16.0_{\pm1.4}$	$72.0_{\pm1.1}$	$66.5_{\pm1.3}$	$67.1_{\pm0.9}$	$81.0_{\pm1.0}$	$73.8_{\pm1.5}$
🔥AFTERIMAGE + ❄LLM + ❄LDM	$\mathbf{0.65}_{\pm.05}$	$\mathbf{0.43}_{\pm.07}$	$\mathbf{0.72}_{\pm.08}$	$\mathbf{0.84}_{\pm.06}$	$\mathbf{0.58}_{\pm.04}$	$\mathbf{0.69}_{\pm.05}$	$\mathbf{11.8}_{\pm0.7}$	$\mathbf{12.4}_{\pm0.8}$	$\mathbf{9.7}_{\pm0.6}$	$\mathbf{12.8}_{\pm1.0}$	$\mathbf{14.0}_{\pm0.9}$	$\mathbf{76.2}_{\pm0.8}$	$\mathbf{72.1}_{\pm1.0}$	$\mathbf{73.8}_{\pm0.7}$	$\mathbf{84.0}_{\pm1.2}$	$\mathbf{81.9}_{\pm0.9}$

3 EXPERIMENTS

We evaluate AFTERIMAGE on brain MRI synthesis (Section 3.1), robot action planning (Section 3.2), and video question answering (Section 3.3) to test three hypotheses.

$\mathcal{H}_1$ Routing the fusion through the AFTERIMAGE code channel preserves or improves task performance, in distribution and under edits and shift.

$\mathcal{H}_2$ AFTERIMAGE code motifs are selectively associated with domain features.

$\mathcal{H}_3$ AFTERIMAGE motifs influence the corresponding domain features in the output.

In the result tables, ❄ marks a frozen component and 🔥 one updated by training.

3.1 BRAIN MRI SYNTHESIS

Setup. We evaluate AFTERIMAGE on the task of synthesizing 3D brain MRI from clinical attributes. The input specifies age, sex, and a four-way diagnostic indicator (EMCI, MCI, LMCI, AD), and the output is a 3D T1-weighted scan matching the attributes. We denote the all-zero diagnostic encoding, a non-Alzheimer's-spectrum label, as CN. The attributes are rendered as a sentence and fused by a frozen Qwen3.5-9B (Qwen Team, 2026) with AFTERIMAGE, and the downstream model is a 3D latent diffusion model (LDM; Rombach et al., 2022; Pinaya et al., 2022) conditioned on the fused representation. We train on 39.9k T1-weighted scans pooled from six public studies and annotated with the three attributes (Appendix E). We use residual routing with $(\alpha, \beta) = (1, 1)$, where the intact backbone stream reaches the decoder alongside the write, and the codes record the adapter's contribution to the conditioning. We compare against three baselines. MLP-LDM encodes the raw attribute vector with a two-layer MLP, LLM-LDM projects the last hidden state of the frozen Qwen without AFTERIMAGE, and Cont. is the continuous variant of AFTERIMAGE, which replaces the discrete codes with continuous slots (Appendices C–G).

Investigating $\mathcal{H}_1$, we compare AFTERIMAGE on 3D brain MRI synthesis and counterfactual editing (CF), where we edit a real scan toward a provided condition. Table 1 reports the comparison in fidelity, measured with FID, and in adherence, measured with the age MAE and sex accuracy (detailed in Appendix F). The results show AFTERIMAGE outperforms all three baselines in fidelity and adherence across synthesis and CF and both CN and impaired cohorts, which is consistent with $\mathcal{H}_1$. The margin over both baselines is largest under CF, where the conditioner must preserve a real scan while changing one attribute. AFTERIMAGE also improves over Cont., which keeps the adapter and replaces the discrete codes with continuous slots, across synthesis and CF, which attributes part of the gain to the quantized bottleneck together with the commitment and coverage terms maintaining the codebook (Appendix D). In addition, Figure 2 shows the qualitative results, where AFTERIMAGE reproduces the enlargement of peri-hippocampal fluid from CN at age 20 to AD at age 90 more closely than baselines and matches the anatomical progression reported for aging and Alzheimer's disease (Jack et al., 1997; Nestor et al., 2008; Fjell & Walhovd, 2010).

Investigating $\mathcal{H}_2$, we extract code motifs and test their selective association with the explicit features provided in the input (diagnosis, age, and sex) and with underlying features extracted from the generated volumes by an automated morphometry tool (Henschel et al., 2020). For each group of conditions, we select the two trigrams (three-code motifs) most specific to the group and evaluate

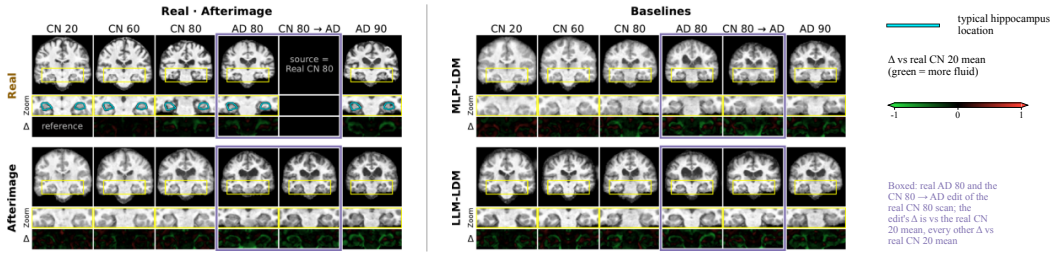


Figure 2: **Generated MRI across age and diagnosis.** Coronal slices from real scans and from each model, ordered from CN at age 20 to AD at age 90. Peri-hippocampal fluid increases along the real progression, and AFTERIMAGE samples track the trend most closely.

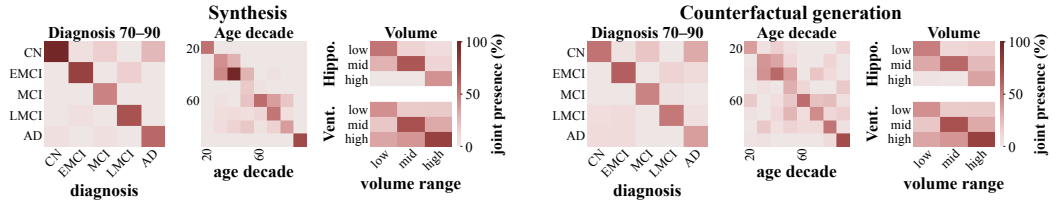


Figure 3: **MRI code signatures.** Percentage of withheld inputs on which both trigrams of a group's signature fire, under synthesis (left) and CF (right). Rows are the groups a signature was selected for, columns are the groups the signature is evaluated on, and off-diagonal cells measure leakage.

the pair on withheld template-and-phrasing combinations (Appendix H). Figure 3 reports, for each selected signature, the percentage of withheld inputs in each evaluation group on which both trigrams fire, where the diagonal measures presence in the signature's own group and the off-diagonal cells measure leakage into other groups. The results show the selected motifs are associated with both the explicit and the underlying features and concentrate on their own group, although the motifs receive no code-level supervision. The associations also hold under CF (Figure 3, right). Against a label-permutation null with false discovery rate (FDR) control, 27 of the 30 group-level associations remain significant, and the exceptions are the 40s and 50s decades and the lowest inferior lateral ventricle tertile (Appendix H, Figure 9). The associations therefore support $\mathcal{H}_2$ under this protocol.

Investigating $\mathcal{H}_3$, we apply **motif substitution**, which overwrites a selected motif and measures the resulting change in the generated brain MRI. A substitution applies only where the assigned codes realize a source trigram, replacing those codes with either a named *target* motif or a *feature-neutral* motif (Appendix H). Figure 4 reports automated volume measurements of the generated scans after substituting the age and diagnosis motifs, against the intact model and real subjects. The results in Figure 4 left and middle show that the measured volumes follow the substituted motif, which indicates an influence of the motifs on the statistics of the generated MRI. Particularly, the 20s or 80s age trigrams hold ventricle volume near the target decade, and the CN or AD diagnosis trigrams hold hippocampal volume near the target category, with a smaller separation than for age. In addition, random codes at the same depths collapse the volumes (Figure 27), and the measured morphometric volumes fall to nearly zero. Besides, Figure 4 right separates the age and morphometric motifs in the 80s conditions, whose record carries an 80s age motif and a high-ventricle motif. Both keep the two motifs; age only and morpho only overwrite the other motif with a feature-neutral motif (a code sequence not selected as any group's signature); neither overwrites both, and age+low replaces the high-ventricle motif with the low-ventricle one. Ventricle volume matches real 80s subjects only under both, falls as motifs are overwritten, and drops substantially toward the low range under age+low, which shows the two motifs contribute separately and the morphometric motif sets the volume level (Appendix H). Motif substitution therefore supplies intervention-based evidence for $\mathcal{H}_3$ on the generated brain MRI.

3.2 ROBOT ACTION PLANNING

Setup. We evaluate AFTERIMAGE on the task of language-conditioned robot manipulation. The input is a text instruction, camera views, and the robot state, and the output is a sequence of actions executed in simulation. The inputs are fused by a frozen SmolVLA (Shukor et al., 2025) or $\pi_{0.5}$ (Physical Intelligence, 2025) vision-language-action (VLA) backbone with AFTERIMAGE, and the

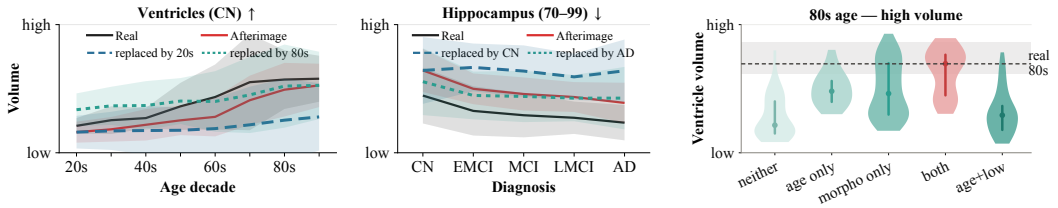


Figure 4: **Motif substitution on brain MRI.** Generated volume after motif overwrites, compared with real subjects. Left and middle replace the age or diagnosis motif with the motif of another decade or category, and right shows 80s ventricle volume by which of the age and high-ventricle motifs remain, and under age+low. Across panels, generated volume follows the substituted motif.

downstream model is the action expert of the backbone, which attends to the keys and values of each layer. We train and evaluate in-distribution on the four LIBERO suites (Liu et al., 2023a) and out-of-distribution on LIBERO-Pro (Zhou et al., 2025), which perturbs the same tasks with paraphrased instructions (Language), relocated objects (Position), and redesigned task logic (Task), and success is the simulator's goal predicate within a fixed step limit (Appendix D.5). We use exclusive routing with $(\alpha, \beta) = (0, 1)$, where the codes mediate what the policy reads at each gated depth, and the layers below the first insertion reach the expert directly (Appendix K). We compare against three baselines. The backbone alone keeps the VLA frozen and fine-tunes its action expert, LoRA adapts the backbone at a matched parameter budget, and Cont. is the continuous-bottleneck variant of AFTERIMAGE with continuous slots in place of the discrete codes (Appendix D.5). A Non-causal variant of AFTERIMAGE additionally removes the causal mask of the gather refiner. The Cont., Non-causal, and AFTERIMAGE arms freeze the expert and train only their adapter.

Investigating $\mathcal{H}_1$ on manipulation, Table 2 reports success rates in distribution and under the Language and Task shifts, and Table 11 adds the Position axis and the Non-causal variant. The results show AFTERIMAGE matches the adapted arms in distribution and outperforms the adapted arms under distribution shift on both backbones. In-distribution, the margins among the adapted arms are small, indicating adapting the VLM accounts for most of the in-distribution gain. Under shift, AFTERIMAGE leads Cont. and LoRA on all but one suite, which attributes the added robustness to the quantized bottleneck together with the codebook terms maintaining it. The gain is largest under paraphrase on SmolVLA and under relocated objects and redesigned logic on $\pi_{0.5}$, consistent with a task identity carried by the codes. The robot results therefore support $\mathcal{H}_1$.

Investigating $\mathcal{H}_2$ on manipulation, we extract task, object, and action-phase motifs from training rollouts and test their selectivity on held-out and paraphrased episodes. A task signature is the pair of trigrams most specific to a task across the insertion depths (Appendix K). Figure 5 reports the percentage of episodes in each group on which a signature fires. The results show task signatures selected on training rollouts remain selective on held-out episodes in distribution and on paraphrased episodes out of distribution. The record can also be read on a single episode as the rollout advances. For this reading, we first translate the simulator's actions into human-readable action phases (Appendix I) and then track which task, object, and phase signatures have fired at each keyframe. Figure 6 shows how the motifs and their sequence over an episode make the fusion behavior behind each prediction readable and localize where the behavior departs from the task (SmolVLA in Appendix K). The task, object, and phase signatures therefore support $\mathcal{H}_2$ on manipulation.

Investigating $\mathcal{H}_3$, we apply **motif substitution** to the task signature in two forms, overwriting the signature with a feature-neutral or other-task motif, and swapping the signatures of two tasks in the same scene. Figure 7 and Table 11 (panel c) report Language success after the overwrites, and Table 12 (Appendix K.4) reports which task is completed after the swap. The results yield three findings. First, overwriting the signature lowers success substantially below the intact policy for both replacements, showing the policy depends on the signature. Second, swapping the signature redirects the policy to the other task, completing task 2 under the task-1 instruction in 25.8% of episodes against 0.0% for a feature-neutral replacement, showing the signature carries a task identity the policy acts on. Third, the redirection persists under the Language shift, where the swap completes task 2 in 21.7% of episodes. Motif substitution therefore supplies intervention-based evidence for $\mathcal{H}_3$ on the behavior of the policy.

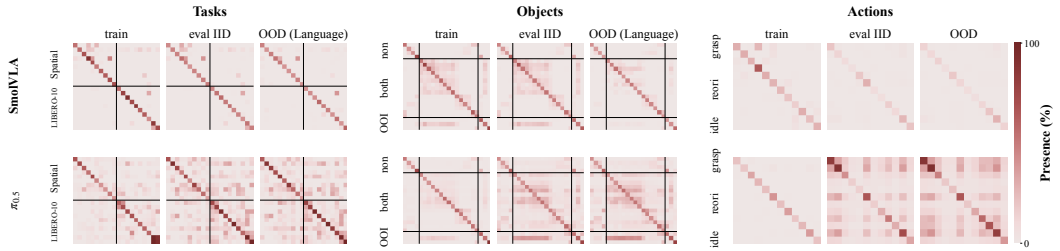


Figure 5: **Robot code signatures on held-out and paraphrased episodes.** Percentage of episodes on which both trigrams of a signature fire. Signatures are selected on training rollouts and evaluated unchanged on held-out and paraphrased episodes; rows are the groups a signature was selected for, and OOI denotes the object of interest. The signatures stay selective on held-out and paraphrased episodes, indicating the motifs capture the task, the scene objects, and the action phases.

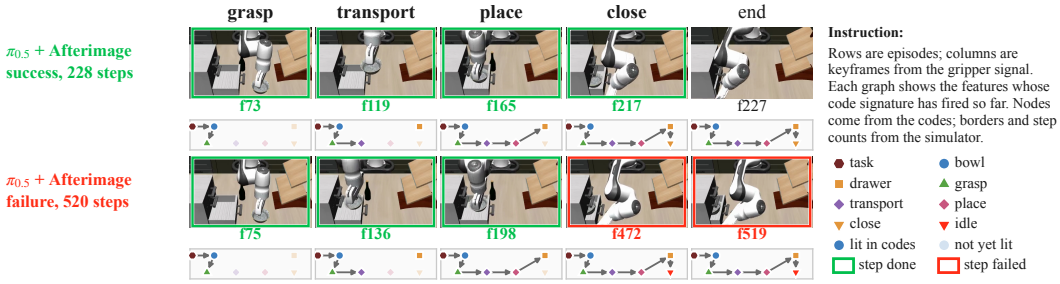


Figure 6: **Rollout progression under AFTERIMAGE ($\pi_{0.5}$).** One successful and one failed episode of the same task. Columns are keyframes from the gripper signal, and the graph below each frame fills in a feature once its signature has fired in the codes. Reading the record as the episode advances shows which task features the policy has acted on, and in the failed episode the point where the signatures stop advancing localizes the error to the closing step.

Table 2: **Robot action planning in distribution and under distribution shift.** Success rate (%) ↑ as mean±std over three runs, and bold marks the best mean. AFTERIMAGE matches the adapted arms in distribution and leads on all but one shifted suite; Table 11 adds the Position suites.

	LIBERO (IID)				Language (OOD)				Task (OOD)			
Arm	Spatial	Object	Goal	LIBERO-10	Spatial	Object	Goal	LIBERO-10	Spatial	Object	Goal	LIBERO-10
SmolVLA (Shukor et al., 2025)	83.2±0.9	91.5±0.5	86.1±1.1	52.4±1.2	43.8±2.4	76.2±1.6	39.1±3.0	28.6±2.2	17.9±2.1	6.4±1.4	13.5±1.9	4.1±1.2
SmolVLA + LoRA	96.0±0.4	98.0±0.3	**98.5**±0.3	**91.8**±0.5	61.5±1.9	83.5±1.4	63.0±2.1	61.6±1.8	6.4±1.5	0.2±0.2	9.6±1.6	**8.5**±1.2
SmolVLA + Cont.	95.3±0.6	**98.4**±0.3	96.9±0.5	89.6±0.8	75.0±1.8	88.8±1.2	55.4±2.3	54.8±1.9	34.5±2.3	2.6±0.8	12.5±1.5	3.7±1.0
SmolVLA + AFTERIMAGE	**96.5**±0.4	97.4±0.2	98.2±0.3	91.0±0.7	**93.8**±1.2	**96.2**±0.8	**69.4**±2.0	**72.6**±1.6	**49.6**±2.3	**9.2**±1.2	**18.8**±1.7	8.4±1.1
$\pi_{0.5}$ (Physical Intelligence, 2025)	98.2±0.5	98.5±0.3	98.3±0.6	97.4±0.8	76.6±2.0	95.1±1.1	41.2±3.0	76.9±1.8	20.6±2.2	6.8±1.5	8.1±1.7	16.4±2.0
$\pi_{0.5}$ + LoRA	95.9±0.5	97.4±0.3	99.0±0.2	97.7±0.4	85.3±1.5	96.1±0.9	79.9±2.0	82.9±1.6	40.4±2.0	23.2±1.9	27.2±2.0	24.8±1.8
$\pi_{0.5}$ + Cont.	95.7±0.5	96.1±0.4	97.4±0.3	98.3±0.4	83.8±1.6	93.8±1.0	80.5±2.1	84.1±1.5	40.6±2.1	23.8±1.9	26.7±1.8	25.0±2.0
$\pi_{0.5}$ + AFTERIMAGE	**98.6**±0.2	**99.3**±0.1	**99.4**±0.2	**98.8**±0.3	**91.4**±1.3	**99.5**±0.4	**89.2**±1.8	**90.6**±1.2	**49.8**±1.9	**33.4**±2.2	**39.0**±2.0	**38.6**±1.8

3.3 VIDEO QUESTION ANSWERING ON CLEVRER

Setup. We evaluate AFTERIMAGE on the task of video question answering. The input is a video and a question about the video, and the output is a short textual answer. The video, represented by 24 frames sampled uniformly across the clip, and the question are fused by a frozen Qwen2.5-VL-7B with AFTERIMAGE inserted into its language stack, and the answer is decoded by the same frozen backbone. We train and evaluate on the open-ended descriptive split of CLEVRER (Yi et al., 2020), where a question is answered by composing objects, their attributes, and the events they take part in (Appendix D.6). We use residual routing as in brain MRI, where the adapter gathers the prompt and visual tokens and the ground-truth answer stays outside the gathered input. We compare against three baselines: the frozen backbone answering directly, LoRA (Hu et al., 2021) at a comparable parameter budget, and Cont., where all arms share backbone, frames, prompt, and decoding.

Investigating $\mathcal{H}_1$ on video, we compare AFTERIMAGE with the three baselines. Table 3 reports accuracy on the open-ended descriptive subset, adds a question-only arm answering without the video, and quotes GPT-4o and Gemini 1.5 Pro from Balazadeh et al. (2024). The results show AFTERIMAGE improves accuracy over the frozen backbone, LoRA, and Cont., which is consistent

Table 3: **CLEVRER video question answering.** Accuracy (%) ↑ as mean±std over three runs for trainable arms. AFTERIMAGE improves over the frozen backbone and both adapted arms.

Model	Acc.	Model	Acc.
Question only	25.3	❄ Qwen2.5-VL-7B (Bai et al., 2025)	71.9
❄ GPT-4o (Hurst et al., 2024)	62.7	❄ Qwen + 🔥 LoRA (0.24%)	83.6$_{\pm 0.3}$
❄ Gemini 1.5 Pro (Gemini Team et al., 2024)	58.6	❄ Qwen + 🔥 Cont. (0.25%)	80.5$_{\pm 0.5}$
		❄ Qwen + 🔥 AFTERIMAGE (0.25%)	**87.9**$_{\pm 0.2}$

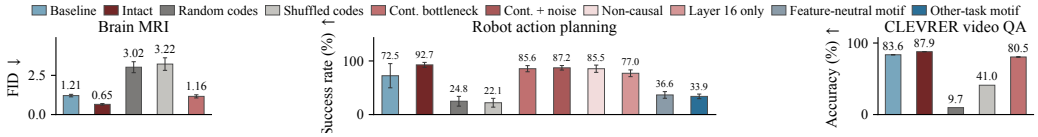


Figure 7: **Ablations of the code channel.** Means over three runs, and the baseline bar is LLM-LDM, the $\pi_{0.5}$ backbone under the Language shift, and LoRA. Across the ablations, comparison results show the benefit of the AFTERIMAGE design in three domains.

with $\mathcal{H}_1$. Cont. falls below LoRA whereas AFTERIMAGE rises above LoRA, and both share the added parameters and the gather–scatter architecture, which attributes the margin to the quantizer.

3.4 ABLATIONS

The ablations test what each design element of the code channel contributes, using two corruptions applied to the codes at inference and four arms retrained with one element removed (Figure 7, Appendix L). The intact AFTERIMAGE performs best in all three domains, and the retrained arms degrade performance less than the corruptions.

The first ablation tests whether the downstream model reads the adapter's write through the codes. Random codes and Shuffled codes, which randomize and permute the codes at inference, both collapse performance in all three domains, which establishes dependence on the code pathway under the tested corruptions, without establishing the meaning of individual codes. The two are similarly damaging on brain MRI and robots, indicating the write depends on which code occupies which slot, whereas on CLEVRER replacing codes is more damaging than permuting the codes, suggesting code identity carries more than order there.

The remaining ablations test whether the elements introduced for the record also affect task performance. Cont. retrains the adapter without the quantizer, and Cont. with noise injection adds Gaussian noise to the slots during training, matching the bottleneck strength without the discrete structure. Cont. worsens FID on brain MRI and lowers accuracy on CLEVRER, and on robots performs comparably in distribution yet trails under all three shifts, while the noise-injected arm narrows the gap on $\pi_{0.5}$ under the Language shift. The quantized bottleneck and the codebook terms maintaining it therefore improve performance most under shift. Non-causal, which removes the causal mask of AFTERIMAGE that orders the slots, lowers success on both policies, most under shift, indicating the ordered refinement contributes to what the policy reads. Restricting AFTERIMAGE to layer 16 of $\pi_{0.5}$ lowers Language success relative to the multi-depth insertion, indicating the earlier depths contribute as well. Across the ablations, the comparisons against the code corruptions and the retrained variants highlight the benefit of the AFTERIMAGE design with discrete codes.

4 CONCLUSION

We proposed AFTERIMAGE, a shared discrete adapter inserted into a frozen multimodal backbone during the fusion process. The adapter records the fusion by a short ordered code sequence, which places an inspectable record on the deployed forward path with under one percent added parameters. Experiments on brain MRI synthesis, robot action planning, and video question answering support three points. First, AFTERIMAGE preserves or improves task performance across domains. Second, AFTERIMAGE's code motifs extracted from the record are associated with domain features. Third, substituting the motifs between features influences the corresponding features in the output. In addition, AFTERIMAGE introduces the capability to read the involved features behind fusion, enabling reading the model behavior through the domain-feature motifs.

AI USE STATEMENT

We used general-purpose large language model assistants, namely Claude (Anthropic), ChatGPT (OpenAI), Gemini (Google), and Perplexity, for three tasks. They helped check the manuscript by cross-checking numbers between the text, tables, and figures and by flagging inconsistent references, captions, and wording. They helped polish the writing by revising the grammar, clarity, and structure of text written by the authors. They also helped write and edit the plotting scripts that render figures from the authors' experimental outputs. The research questions, method, experimental design, and experiments are the authors' own, and the authors reviewed all AI-assisted text, code, and figures and take full responsibility for the content of this paper.

ETHICS STATEMENT

This work uses only publicly available, de-identified data and synthetic or simulated environments, and no new data were collected from human participants. The brain MRI corpus pools six studies (ABCD, ADNI, AIBL, PPMI, HCP-Aging, and HCP-Development), each accessed under its data use agreement. The robot experiments run in simulation on the LIBERO and LIBERO-Pro benchmarks, and CLEVRER consists of synthetic videos. Generated brain scans are research artifacts and are not suitable for clinical diagnosis or for training clinical systems without independent validation. We evaluate adherence to the conditioning attributes with automated evaluators only, so clinical use would additionally require reader studies and validation against clinical endpoints. The model can also inherit biases in the data. In particular, CN denotes the all-zero diagnostic encoding, a non-Alzheimer's-spectrum label rather than verified health (Appendix E), diagnostic labels come from a subset of studies, young adults are under-represented, and generated scans skew toward one sex, so conclusions may not hold for under-represented groups. Synthetic scans could also be mistaken for real patient data, so any released samples should be clearly marked as synthetic.

REPRODUCIBILITY STATEMENT

Section 2 specifies the model and the training objective (Equation 6), and Appendix C gives every layer, the numerical settings, and one full training step (Algorithm 1). Appendix D lists hyperparameters and the codebook maintenance procedure, and Appendix D.2 gives, for every experiment, the backbone and the layers followed by an adapter. Appendices D.4, D.5, and D.6 give the training and evaluation settings of every domain, including what each arm trains and freezes; every trainable arm is trained with three seeds and reported as mean$\pm$std. Appendix E describes the six brain MRI source studies, the diagnostic-label encoding, the condition vector, the preprocessing, and the subject-disjoint splits, and Appendix F defines every metric. Appendix H gives the complete procedure and thresholds for extracting trigram signatures, and Appendix J states how the qualitative samples were selected. Appendix K reports every LIBERO and LIBERO-Pro suite, and Appendix L reports each ablation on every metric.

REFERENCES

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Vahid Balazadeh, Mohammadmehdi Ataei, Hyunmin Cheong, Amir Hosein Khasahmadi, and Rahul G. Krishnan. Physics context builders: A modular framework for physical reasoning in vision-language models. *arXiv preprint arXiv:2412.08619*, 2024.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, et al. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.

Zixuan Chen, Chongkai Gao, Lin Shao, Jieqi Shi, Jing Huo, and Yang Gao. ManiLong-Shot: Interaction-aware one-shot imitation learning for long-horizon manipulation. *arXiv preprint arXiv:2512.16302*, 2025.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*, 2023.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, 2023.

Anders M. Fjell and Kristine B. Walhovd. Structural brain changes in aging: courses, causes and cognitive consequences. *Reviews in the Neurosciences*, 21(3):187–221, 2010.

Ankur Garg, Xuemin Yu, Hassan Sajjad, and Samira Ebrahimi Kahou. Cross-layer discrete concept discovery for interpreting language models. *arXiv preprint arXiv:2506.20040*, 2025.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Leonie Henschel, Sailesh Conjeti, Santiago Estrada, Kersten Diers, Bruce Fischl, and Martin Reuter. FastSurfer – a fast and accurate deep learning based neuroimaging pipeline. *NeuroImage*, 219: 117012, 2020.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning (ICML)*, 2019.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2021.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Clifford R. Jack, Ronald C. Petersen, Yue Cheng Xu, Stephen C. Waring, Peter C. O'Brien, Eric G. Tangalos, Glenn E. Smith, Robert J. Ivnik, and Emre Kokmen. Medial temporal atrophy on MRI in normal aging and very mild Alzheimer's disease. *Neurology*, 49(3):786–794, 1997.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and João Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, 2021.

Sarthak Jain and Byron C. Wallace. Attention is not explanation. In *NAACL-HLT*, 2019.

Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *IEEE/CVF International Conference on Computer Vision*, 2019.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning (ICML)*, 2020.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Yutai Li, Shaohui Peng, Jiaming Guo, Di Huang, Zihao Zhang, Yuxuan Guo, Yunkai Gao, Siming Lan, Ling Li, Xing Hu, and Yunji Chen. PrimitiveVLA: Learning reusable motion primitives for efficient and generalizable robotic manipulation. *arXiv preprint arXiv:2605.28634*, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023b.

Sean M. Nestor, Raul Rupsingh, Michael Borrie, Matthew Smith, Vittorio Accomazzi, Jennie L. Wells, Jennifer Fogarty, and Robert Bartha. Ventricular enlargement as a possible measure of Alzheimer's disease progression validated using the Alzheimer's disease neuroimaging initiative database. *Brain*, 131(9):2443–2454, 2008.

Physical Intelligence. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Walter H. L. Pinaya, Petru-Daniel Tudosiu, Jessica Dafflon, Pedro F. da Costa, Virginia Fernandez, Parashkev Nachev, Sebastien Ourselin, and M. Jorge Cardoso. Brain imaging generation with latent diffusion models. In *MICCAI Workshop on Deep Generative Models*, 2022.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5), 2019.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Alex Tamkin, Mohammad Taufeeque, and Noah D. Goodman. Codebook features: Sparse and discrete interpretability for neural networks. *arXiv preprint arXiv:2310.17230*, 2023.

Frederik Träuble, Anirudh Goyal, Nasim Rahaman, Michael C. Mozer, Klaus Greff, Yoshua Bengio, and Bernhard Schölkopf. Discrete key-value bottleneck. In *International Conference on Machine Learning (ICML)*, 2023.

Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, S. M. Ali Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Sarah Wiegreffe and Yuval Pinter. Attention is not not explanation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: Collision events for video representation and reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

A LIMITATIONS AND FUTURE WORK

Limitations. We presented AFTERIMAGE, a discrete adapter that records its contribution to a frozen backbone's fusion as a short sequence of discrete codes, matches or improves task performance across the three evaluated domains, and yields motifs associated with domain features on held-out data. We identify, however, four boundaries of the present evidence, concerning the scope of the record, the comparisons, the signatures, and the settings evaluated.

- **Scope of the record.** The code sequence determines the adapter's write rather than the backbone's output, so code substitution is a well-defined intervention on the adapter's contribution. Under residual routing (brain MRI and CLEVRER) the intact backbone stream also reaches the downstream model alongside the write, and the adherence remaining after a feature-neutral overwrite (Table 10) measures that margin, whereas under exclusive routing (robots) the next layer's attention at an insertion reads the write alone.
- **Comparisons.** Cont. removes the quantizer together with the commitment and coverage terms defined only for a codebook, and on robots continuous codes and LoRA largely match AFTERIMAGE in distribution, so the in-distribution gains are best read as gains from adapting the VLM.
- **Signatures.** Naming a motif requires candidate features supplied from outside the record, and on LIBERO each scene object belongs to one task, so object signatures are not separable from task identity.
- **Settings evaluated.** The codebook size V and the insertion depths are set before training, and the experiments cover three domains and two VLA backbones.

Future work. The record opens a way to read what a fusion carries in domains where the relevant features are not known in advance. Mining recurring motifs without supervision and naming the motifs with domain experts or a language model would remove the need for a predefined list of features, and unseen object–goal combinations together with phase-labelled evaluation episodes would test whether the codes compose features rather than index a task. Learning the vocabulary size during training, by growing, merging, or pruning codes, would let the resolution of the record follow the domain. Larger or continually adapted backbones, and a comparison of exclusive and residual routing under matched conditions within one domain, would map how far the code structure persists. The same design applies wherever heterogeneous inputs are fused inside a frozen backbone, for example pathology slides with genomic profiles, or medical images with electronic health records, where reading the fused representation would give a per-input account of the domain features entering a downstream decision.

B EXTENDED RELATED WORK

The related work follows the argument of Section 1, moving from the use of pretrained models as fusers, through the difficulty of inspecting the fused state and the methods examining the fused state, to the lines of work AFTERIMAGE builds on directly.

Pretrained models as a fusion mechanism. A frozen pretrained backbone has become a standard site for combining heterogeneous sources. In vision-language systems, a visual encoder is attached to a frozen language model, which relates the two streams. Tsimpoukelli et al. (2021) show a frozen language model conditioned on projected image embeddings acquires multimodal few-shot ability, Flamingo interleaves gated cross-attention into frozen layers (Alayrac et al., 2022), BLIP-2 inserts a querying transformer between a frozen image encoder and a frozen language model (Li et al., 2023), and LLaVA projects visual features into the language embedding space (Liu et al., 2023b). Generative pipelines apply the same pattern on the conditioning path, encoding a verbalized description with a frozen text encoder and cross-attending to the encoder's hidden states (Rombach et al., 2022), with extensions to large frozen language encoders (Saharia et al., 2022) and to attribute-conditioned synthesis of 3D medical volumes (Pinaya et al., 2022). Embodied policies, including VLA models, fuse language, vision, and state through a pretrained backbone before emitting actions (Driess et al., 2023; Brohan et al., 2023), and joint embedding models serve as general relational encoders (Radford et al., 2021). Across the pipelines, the backbone is expected to relate the sources rather than only extract features, and the downstream model receives a hidden state selected by layer

index. AFTERIMAGE occupies the same position and emits, alongside the continuous representation, a discrete record intended to be read. The closest baseline is therefore the same design without the quantizer, reported as Cont. in the main tables, alongside LoRA on CLEVRER and on both VLA backbones.

Inspecting the fused state. The case for models constrained to be interpretable in high-stakes settings, rather than explained afterwards, is long-standing (Rudin, 2019), and the case applies directly to fusion. The fused state is dense and unlabelled, so the state alone does not indicate which sources were retained, how the sources interacted, or whether the backbone's prior was used. Fusion errors are consequently hard to detect, as a fusion dropping an attribute can still yield a plausible output, and expert review is least able to catch plausible errors at scale.

Explanation by generated text. One family of methods asks the model to state its reasoning in natural language, which expresses the explanation in terms a practitioner can read. The explanation is produced after the computation fusing the sources, so the text reports on the computation rather than recording the computation, and the report can diverge from what determined the output. Turpin et al. (2023) show stated reasons can omit features determining the answer, and Lanham et al. (2023) find chain-of-thought can be perturbed or truncated while the conclusion stands. Decoding is also stochastic, so one hidden state can map to several different explanations. AFTERIMAGE reverses the ordering, as the codes are emitted during the fusion and form the channel the downstream model reads. Corrupting the codes at inference therefore tests the downstream model's dependence on the code pathway directly (Section 3.4).

Post-hoc analysis of internal representations. A second family recovers structure from the trained network after training. Linear probes read attributes from intermediate activations (Alain & Bengio, 2016; Belinkov, 2022), sparse autoencoders and dictionary learning decompose activations into interpretable directions (Cunningham et al., 2023; Bricken et al., 2023), and attention weights are widely used for attribution, although their explanatory status is debated (Jain & Wallace, 2019; Wiegreffe & Pinter, 2019). The methods are grounded in the model's computation and provide complementary views of learned representations, whereas the resulting explanatory objects are constructed after training and typically characterize individual directions rather than how sources combine. A probe reports whether an attribute is decodable, and a dictionary reports which directions are active. In AFTERIMAGE, the vocabulary is instead learned jointly with the task, so the object being read is part of the trained system rather than recovered from the system.

Interpretable-by-design bottlenecks. Concept bottleneck models route prediction through a layer of human-specified concepts, placing the interpretable object inside the model (Koh et al., 2020). Three differences separate the present setting. Concept bottlenecks are mainly used for discriminative prediction, whereas the AFTERIMAGE bottleneck sits on a fusion pathway, which allows the effect of a code on generated content to be measured rather than only the effect on a label. Concept bottlenecks also assume a pre-specified, annotated concept set, whereas the AFTERIMAGE vocabulary is learned with the downstream task and analyzed against domain features after training. The third difference is representational, as a concept bottleneck exposes an unordered activation vector, whereas the AFTERIMAGE codes are refined left to right and decoded with position embeddings, so a fused input yields a sequence. On brain MRI and robot action planning, permuting code order at inference is about as damaging as replacing the codes at random (Section 3.4).

Discrete representation learning. Vector quantization with a learned codebook (van den Oord et al., 2017) underpins the bottleneck, and codebook collapse is a well-documented difficulty, commonly mitigated by usage-entropy terms and dead-code revival. AFTERIMAGE applies the quantizer to a fusion pathway with few slots per input rather than to compressing the data itself.

Codebook features (Tamkin et al., 2023) are the closest prior work. The method finetunes a language model with a vector-quantization codebook after every attention head or MLP block, replacing each head or block output at every token with the sum of the top-k most similar code vectors before the output enters the residual stream, and reads and activates the codes to interpret and steer the model. AFTERIMAGE shares the central commitment to a readable discrete code on the computation path of the deployed network rather than a surrogate fitted to activations. The designs differ in four respects. Codebook features train the codebooks jointly with every backbone weight, whereas AFTERIMAGE

keeps every pretrained weight frozen and trains 0.20–0.71% added parameters. Codebook features use a separate codebook for each layer and head, so a code index is specific to its site, whereas AFTERIMAGE shares one codebook across depths, so an index denotes the same entry at every insertion and a motif can be followed through the backbone. Codebook features write an unordered sum of k codes for every token at every layer, whereas AFTERIMAGE writes an ordered sequence of K slots per input at a few depths, which keeps a record short enough to read per example and lets order carry information (Section 3.4). Codebook features are studied on text-only language modelling, whereas AFTERIMAGE places the bottleneck on a fusion pathway whose output a separate downstream model consumes. Träuble et al. (2023) route representations through a discrete key–value bottleneck to localize updates and improve robustness under distribution shift, rather than to expose a vocabulary for reading. Garg et al. (2025) also place a discrete bottleneck on the residual stream and read the bottleneck across layers, whereas their codebook is learned under a separate concept-discovery objective and the AFTERIMAGE codebook is learned under the downstream task consuming the fusion. The shared codebook and the ordered slots allow a motif to be compared across depths and read in the order the motif is composed, as used in Sections 2.2 and 2.4.

Parameter-efficient adaptation. Adapters and low-rank updates specialize a frozen backbone at small parameter cost (Houlsby et al., 2019; Hu et al., 2021). AFTERIMAGE is parameter-efficient in the same sense and leaves the backbone unchanged, so the backbone's prior is preserved by construction. The difference lies in legibility, as a low-rank weight update specializes the model without exposing which knowledge is used, whereas a discrete vocabulary can be enumerated, counted, and compared across inputs and depths.

Language-conditioned robot action planning. LIBERO (Liu et al., 2023a) evaluates instruction-following manipulation across suites varying spatial reference, object reference, goal specification, and long-horizon composition. LIBERO instructions are compositional rather than drawn from a fixed label set, which makes the benchmark suitable for testing whether code motifs track task structure beyond the identity of individual inputs.

C ARCHITECTURE DETAILS

This appendix records the layer-level details of the adapter, complementing the component definitions and the objective given in Section 2.

Latent-query lineage. The gather cross-attention is the latent-query resampler of Perceiver and of the Q-Former of BLIP-2 (Jaegle et al., 2021; Li et al., 2023), adopted unchanged as the interface between the residual stream and the slots. Whereas a Q-Former sits once at the encoder–decoder interface and emits continuous queries, $\mathcal{A}$ is reused inside the backbone at several depths and emits discrete codes. The K queries are distinct learned parameters, which gives slot j a fixed identity with or without the causal mask. The causal mask adds an ordering constraint, under which z_j depends only on $z_1, \ldots, z_j$ among the slots. The codes therefore form a left-to-right code sequence rather than K independently filled positions.

Layers. The adapter has no depth-specific parameters. One $\mathcal{A}$, one codebook, and one pair $W_\downarrow, W_\uparrow$ serve every insertion in $\mathcal{S}$, and the routing configuration (α, β) is fixed within an experiment. The projections $W_\downarrow$ and $W_\uparrow$ are bias-free and reduce to the identity when $d_a = d$. The gather step adds a learned position embedding to the K queries and to the L input positions, applies layer norms to the queries and to the keys and values, and normalizes the attention output. The subsequent slot refinement uses pre-norm transformer encoder layers with a causal mask over slots. Dropout is disabled in the refinement, and training and inference therefore compute the same codebook IDs. The scatter step uses L learned position queries U, which attend to the quantized slots tagged with slot-position embeddings P_{slot} and are followed by a pre-norm feed-forward block. Each of the cross-attention and feed-forward blocks has a residual connection and is followed by a layer norm. In our implementation, $\mathcal{T}$ reads the output of F through Pool, a layer-normed cross-attention from N_q learned queries followed by a linear map to the input width of $\mathcal{T}$.

Algorithm 1 One training step of AFTERIMAGE. The frozen backbone runs without gradients up to the first insertion depth. At every depth in $\mathcal{S}$, the same adapter gathers the residual stream into K ordered slots, quantizes each slot against the shared codebook, scatters the quantized slots back to length L, and routes the write through (α, β). One module, one codebook, and one pair $W_\downarrow, W_\uparrow$ serve every insertion, and adding a depth therefore adds computation rather than parameters. The commitment and coverage terms accumulate over the depths and join the task loss. Gradient descent updates the trainable parameters through the straight-through estimator, the codebook entries follow an EMA update, and every pretrained weight stays fixed.

1: $h \leftarrow$ embed the inputs and run the frozen layers $f_1, \ldots, f_{\min \mathcal{S}}$ without gradients
2: **for** $i \in \mathcal{S}$ in increasing order **do**
3: run the frozen layers up to f_i, then gather and quantize $u^{(i)} = h^{(i)} W_\downarrow$ into $\tilde{Z}$ and $\kappa^{(i)}$ ▷ Eqs. 2–3
4: scatter $\tilde{Z}$ into $a^{(i)}$, set $w^{(i)} \leftarrow a^{(i)} W_\uparrow$ and $\tilde{h}^{(i)} \leftarrow \alpha h^{(i)} + \beta w^{(i)}$, and accumulate $\lambda_{\text{commit}} \mathcal{L}^{(i)}_{\text{commit}} + \lambda_{\text{cov}} \mathcal{L}^{(i)}_{\text{cov}}$ ▷ Eqs. 5, 4
5: **end for**
6: run the remaining frozen layers, pass the output of F to $\mathcal{T}$, and evaluate $\mathcal{L}$ ▷ Eq. 6
7: update the trainable parameters by gradient descent and the codes by EMA

D IMPLEMENTATION DETAILS

This appendix gives the settings needed to reproduce each experiment, namely the vector-quantization terms, the per-experiment configuration, the parameter counts, and the training and evaluation setup of brain MRI synthesis, robot action planning, and CLEVRER question answering.

Training length and checkpoint selection. Every epoch count in this appendix is a ceiling rather than the point at which training stopped improving. In all three tasks, each arm is trained until its validation objective stops improving, and the reported checkpoint is the one with the best validation score. The selected checkpoint falls inside the ceiling for every arm, which indicates that training length does not limit the comparisons in this paper.

D.1 VECTOR-QUANTIZATION TERMS

This subsection gives the form of the standard commitment and coverage terms, optimized jointly with the task objective in Section 2.3, together with the codebook mechanics omitted from the main text. Both terms are summed over the inserted adapters and weighted by λ_{commit} and λ_{cov} in Equation 6. Neither weight is the routing coefficient β of Equation 5.

$$\mathcal{L}_{\text{commit}} = \frac{1}{K} \sum_j \left\| \hat{z}_j - \text{sg}[\hat{e}_{k_j}] \right\|_2^2, \qquad \mathcal{L}_{\text{cov}} = \frac{\mathbb{E}_z[H(p_z)] - H(\mathbb{E}_z[p_z])}{\log V}, \tag{7}$$
$$p_z = \text{softmax}\big(\hat{z}^\top \hat{e}_{1:V} / \nu\big).$$

The commitment term keeps each slot close to its assigned entry. The codebook itself receives no gradient from the commitment term and is instead updated by an exponential moving average of the slots assigned to each entry (van den Oord et al., 2017; Razavi et al., 2019), at the decay given with Table 4. The coverage term is the negative mutual information between slots and entries under the soft assignments p_z, normalized by $\log V$ for comparability across codebook sizes. Minimizing the coverage term maximizes the mutual information, which counters collapse onto a few entries (Ji et al., 2019). The marginal entropy $H(\mathbb{E}_z[p_z])$ is estimated over the slots in the current batch, and the strength of the coverage term therefore depends mildly on batch size.

Assignment. Each slot takes its nearest entry independently of the other slots. An ID therefore denotes the same codebook entry wherever it appears, which makes a motif directly comparable across examples and depths.

Similarity and temperature. Assignment, the commitment term, and p_z operate on ℓ_2-normalized slots and entries. The geometry is therefore cosine rather than Euclidean, and the assignment is invariant to slot magnitude. The un-normalized slot z enters no term, since Equation 3

carries $\hat{e}_{k_j}$ forward and the scatter step reads the normalized entries. The assignment temperature ν enters only through p_z in $\mathcal{L}_{\text{cov}}$ and not through the arg-max of Equation 3. The temperature therefore sets how sharply coverage is enforced and does not enter the assignment itself.

Vocabulary maintenance. Two maintenance rules keep the codebook spread over distinct, used directions. Codes are initialized by farthest-point sampling over the slots of the first training batch, which spreads the initial codebook over distinct directions. Every 250 steps, a code is reseeded if its EMA usage falls below 0.1 of the mean usage, or if the code is the less-used member of a code pair whose cosine similarity exceeds 0.9. Reseeds are the slots of the current batch least similar to any code, perturbed by small Gaussian jitter. Drawing reseeds from the least similar slots places them in under-covered regions and avoids duplicating an existing code.

D.2 Per-experiment configuration

Table 4 lists, for every experiment, the frozen backbone, the selected layers $\mathcal{S}$, the bottleneck sizes (K, V, d_a), the routing coefficients, and the quantization hyperparameters. Only a few layers are followed by an adapter, and the remaining backbone layers are unchanged. Indices are 1-based, and $i \in \mathcal{S}$ denotes an adapter following the i-th layer f_i. The MRI and CLEVRER settings share $K{=}16$, $V{=}100$, $d_a{=}512$; the robotics settings share $K{=}8$ and $V{=}64$, and set d_a to 128 on SmolVLA and 512 on $\pi_{0.5}$. The vector-quantization settings are shared across experiments. The codebook EMA decay is 0.99, the assignment temperature is $\nu{=}0.5$, the commitment and coverage weights are $\lambda_{\text{commit}}{=}0.25$ and $\lambda_{\text{cov}}{=}1.0$, and the gather refiner uses 2 layers with 4 attention heads. A code is reseeded every 250 steps if its EMA usage falls below 0.1 of the mean usage or if the code is the less-used member of a pair whose cosine similarity exceeds 0.9.

Table 4: Per-experiment configuration. B: backbone layers; d: backbone width; $\mathcal{S}$: layers followed by an adapter (1-based); K: ordered slots; V: codebook size; d_a: adapter width; (α, β): routing coefficients. Vector-quantization settings are shared across experiments and given above.

Experiment	Backbone	B	d	$\mathcal{S}$	$\lvert\mathcal{S}\rvert$	K	V	d_a	(α, β), routing
Brain MRI	Qwen3.5-9B (Qwen Team, 2026)	32	4096	$\{4, 8, 12, 16, 20, 24, 28, 31\}$	8	16	100	512	$(1, 1)$, residual
CLEVRER	Qwen2.5-VL-7B (Bai et al., 2025)	28	3584	$\{4, 8, 12, 16, 20, 24, 28\}$	7	16	100	512	$(1, 1)$, residual
LIBERO	SmolVLA (Shukor et al., 2025)	8	960	$\{2, 4, 6, 8\}$	4	8	64	128	$(0, 1)$, exclusive
LIBERO	$\pi_{0.5}$ (Physical Intelligence, 2025)	18	2048	$\{4, 8, 12, 16\}$	4	8	64	512	$(0, 1)$, exclusive

D.3 Model size and trainable parameters

Table 5 records the size of every component and makes the parameter share quoted in the abstract and in Section 1 checkable. *Share* is the number of AfterImage parameters as a percentage of backbone parameters. The count covers the adapter $\mathcal{A}$, the projections $W_{\downarrow}$ and $W_{\uparrow}$, and the codebook. The task-specific model $\mathcal{T}$ is trained in some domains and frozen in others, and the table therefore reports $\mathcal{T}$ separately and excludes $\mathcal{T}$ from the share.

Table 5: **Model size and trainable parameters.** Share = AfterImage params / backbone params, covering the adapter, the projections $W_{\downarrow}, W_{\uparrow}$, and the codebook. The adapter, projections, and codebook are shared across insertion depths, and the share is therefore set by K, V, the adapter width, and the sequence length L spanned by the scatter queries rather than by the number of depths. The backbone column counts every pretrained weight, and the AfterImage arms keep the whole backbone frozen. The task-specific model $\mathcal{T}$ is listed separately and excluded from the share. Every arm of a domain reads through the same $\mathcal{T}$, and the arms differ in the conditioner trained in front of the frozen stack.

		Backbone		AfterImage		Task model $\mathcal{T}$	
Experiment	Backbone	Params	State	Params	Share (%)	Params	State
Brain MRI	Qwen3.5-9B (Qwen Team, 2026)	9.65B	❄	19.6M	0.20	39.1M (3D UNet)	❄
CLEVRER	Qwen2.5-VL-7B (Bai et al., 2025)	8.29B	❄	20.6M	0.25	—	❄
LIBERO ($\lvert\mathcal{S}\rvert = 4$)	SmolVLA (Shukor et al., 2025)	271.5M	❄	1.94M	0.71	50.8M (action expert)	❄
LIBERO ($\lvert\mathcal{S}\rvert = 4$)	$\pi_{0.5}$ (Physical Intelligence, 2025)	3.45B	❄	20.8M	0.60	693.4M (action expert)	❄

D.4 BRAIN MRI SETUP

Diffusion decoder. The generator is a latent diffusion model whose denoiser receives the fused representation through cross-attention. Volumes are single-channel $160 \times 192 \times 176$ at $1\,\mathrm{mm}$ (Appendix E). A MONAI `AutoencoderKL` with channels $[64, 128, 256]$, two residual blocks per level, and no attention encodes each volume into a $32 \times 40 \times 48 \times 44$ latent, that is, 32 channels at a $4\times$ spatial reduction. The latent is rescaled by 0.9957, the reciprocal of the standard deviation of the first training batch. The denoiser is a MONAI `DiffusionModelUNet` with channels $[128, 128, 256]$, one residual block per level, attention at the second and third levels with 64 head channels, and cross-attention of width 256. The fused representation enters the denoiser through the cross-attention as $T = 2$ tokens. Training uses a DDPM scaled-linear schedule with β from 0.0015 to 0.0195 over 1000 timesteps. Sampling runs the same scheduler for all 1000 steps and uses neither strided nor DDIM sampling nor classifier-free guidance. The backbone runs once per generated sample, and one code sequence per insertion depth therefore conditions all 1000 denoising steps of the sample.

Training. The MRI protocol makes the trained conditioner the only difference between the two language-model arms. The Qwen3.5-9B backbone and the autoencoder are frozen throughout, and every arm optimizes a noise-prediction (ϵ) MSE objective. LLM-LDM first trains its pooler together with the 3D UNet for 150 epochs. Both language-model arms then branch from the resulting checkpoint and continue for up to 50 further epochs with the decoder frozen. In the second stage, LLM-LDM keeps training its pooler, whereas AFTERIMAGE inherits the pooler frozen, trains only the inserted adapter, and adds the vector-quantization terms of Appendix D.1 to the same objective. The two arms therefore share a branch point, a decoder, and the second-stage epoch budget, and differ only in the trained conditioner. MLP-LDM uses no language model and trains its conditioner together with the UNet for up to 200 epochs. Every stage runs the same training script and configuration. The optimizer is AdamW with $\beta = (0.9, 0.999)$, $\varepsilon = 10^{-8}$, and weight decay 10^{-2}. The learning rate is 2.5×10^{-6} per process and scales with the number of processes, giving 10^{-5} on the four H100 GPUs used for every arm. The batch size is 1 per GPU, and training uses neither gradient accumulation nor gradient clipping. The stages differ only in which parameters enter the optimizer. MLP-LDM trains a small conditioner and the UNet, whereas in the shared second stage LLM-LDM trains its 68.2M pooler and AFTERIMAGE its own 19.6M adapter and projections, both against the same frozen decoder. The continuous-bottleneck ablation of Section 3.4 shares the second stage and the checkpoint rule below. Each arm is trained three times with different seeds, and the reported values are the mean$\pm$std over the three runs. The random-code and shuffled-code conditions of Section 3.4 likewise average three runs.

Reported checkpoints. Every arm is selected by the same rule, the lowest validation diffusion reconstruction loss. MLP-LDM is selected at epoch 187 of 200; in the shared second stage, LLM-LDM is selected at epoch 46 of 50 and AFTERIMAGE at epoch 29 of the same 50. AFTERIMAGE is therefore reported after fewer epochs of conditioner training than LLM-LDM, from the same branch point, against the same frozen decoder, and with a smaller trained component. Longer training thus does not account for the difference between the two arms.

D.5 ROBOT SETUP

Training setup. The robot protocol, applied to both VLA backbones, gives every arm the same starting policy and the same second-stage epoch budget. Both arms use the LIBERO demonstrations, which comprise $1{,}654$ episodes and $262{,}335$ frames after the official-demo validation filter; the filter drops 39 LIBERO-10 episodes. Training proceeds in two stages. In the first stage, the vision encoder and the VLM are frozen, and the action expert is trained for 20 epochs to convergence. Every arm shares the resulting checkpoint. In the second stage, each arm continues from the shared checkpoint for a further 20 epochs. The baseline arm keeps training the action expert, whereas the AFTERIMAGE arm freezes the checkpoint entirely (vision encoder, VLM, and action expert), inserts the shared adapter into the VLM at the depths of Table 4 under exclusive routing, and trains the adapter alone. Every arm therefore starts from the same policy weights and trains for the same number of additional epochs. Additional training of the policy thus does not explain the reported gains. Over the second-stage epochs, the backbone arm continues training the action expert, whereas

AFTERIMAGE trains the inserted adapter instead and leaves every policy weight fixed; both arms are trained to convergence.

LoRA baseline. The robot LoRA arm adapts the whole action expert together with the query and value projections of the VLM and the projection layers, for 101 target modules at rank 13, giving 2,279,056 trainable parameters on SmolVLA. The LoRA arm branches from the same LIBERO-finetuned checkpoint as every other arm.

Continuous bottleneck and shared protocol. The Cont. arm compares discrete with continuous codes while holding the rest of the AFTERIMAGE arm fixed. Cont. uses the same module at the same insertion depths, with the same routing and the same parameter count up to the dropped codebook. The slots reach the scatter block unquantized rather than mapped to codebook entries, and training follows the two-stage protocol of the AFTERIMAGE arm unchanged. Cont. keeps the gather and scatter blocks, the causal refinement over slots, and the adapter width d_a. Cont. drops the codebook together with the commitment and coverage terms of Equation 6, since both terms are defined on the codebook and have no form without one. The Cont. objective is therefore the task loss alone. The gap between Cont. and AFTERIMAGE thus measures the quantized bottleneck as trained, together with the commitment and coverage terms maintaining the bottleneck, rather than discreteness in isolation. For every arm we report the second-stage checkpoint with the best validation score, following the checkpoint rule of Appendix D. Each arm is trained three times with different seeds, and the reported success rates are the mean$\pm$std over the three runs. For $\pi_{0.5}$, the first stage starts from the public checkpoint and retrains the action expert on the LIBERO demonstrations as for SmolVLA. Every reported $\pi_{0.5}$ result therefore comes from the retrained checkpoint rather than from the released one. Both arms see the same demonstrations, environment configuration, and observation and action spaces. Both are trained on two NVIDIA L40S GPUs and evaluated with the per-episode protocol described below. Table 6 lists the settings of both stages for each backbone.

Table 6: **Robot training settings.** All arms of a backbone share the first stage and its checkpoint, and the arms differ only in what the second stage updates. In the second stage, each arm continues from the shared checkpoint for the same number of epochs on the same data. The baseline arm trains the action expert further, whereas the Cont. and AFTERIMAGE arms freeze the checkpoint entirely and train only their added parameters. Optimizer, schedule, batch size, and action-chunk settings are identical across the arms of a backbone.

	SmolVLA	$\pi_{0.5}$
Optimizer	AdamW	AdamW
Weight decay	0.01	1×10^{-10}
Gradient clipping	1.0	10.0
Learning-rate floor	2.5×10^{-6}	2.5×10^{-6}
Batch size	32	32
Stage 1 epochs (shared, action expert)	20	20
Stage 2 epochs (per arm)	20	20

Evaluation protocol. The evaluation protocol holds episodes, seeds, and step limits fixed across arms and backbones. Every episode is evaluated in its own process, and each arm is therefore reset from the same seed and starts from the same layout. Both backbones use the protocol of Table 7. The four LIBERO suites, crossed with the in-distribution setting and the three LIBERO-Pro shifts, give sixteen conditions, and every arm runs the same number of episodes in each condition. The policy predicts a chunk of actions at each decision and executes a prefix of the chunk before re-planning. An episode counts as a success when the simulator's goal predicate for its task is satisfied before the step limit, and as a failure otherwise. A failed episode therefore runs to the limit, and its length is uninformative. The limit is LIBERO's default in distribution (280 steps on Spatial and Object, 300 on Goal, and 520 on LIBERO-10) and 500 steps on every LIBERO-Pro suite. The protocol applies to every reported success rate, in Table 2 and Table 11, and to the episodes shown in Figures 6 and 20.

Table 7: **Robot evaluation protocol.** The protocol is identical for SmolVLA and $\pi_{0.5}$ and for every arm of each backbone, and every success rate in Tables 2 and 11 therefore rests on the same number of episodes. Ten tasks per suite at fifty episodes per task give 500 episodes per condition. The sixteen conditions cover four suites under in-distribution evaluation and under each of the three LIBERO-Pro shifts, for 8,000 episodes per arm. In every arm and condition, each decision predicts a chunk of fifty actions, of which ten are executed before the next decision.

Tasks per suite	10
Episodes per task	50
Episodes per condition	500
Conditions (4 suites $\times$ {IID, Language, Position, Task})	16
Total episodes per arm	8,000
Predicted action chunk	50
Executed action steps per decision	10

D.6 CLEVRER SETUP

The CLEVRER setup holds the backbone, prompt, and decoding fixed across arms and varies only the trained adaptation. The backbone is a frozen Qwen2.5-VL-7B-Instruct whose vision encoder, language model, and `lm_head` remain unchanged. Each video is represented by 24 frames sampled uniformly over the full clip at native resolution. Every arm receives the same prompt, consisting of the video, the question, and the instruction *"Answer with a single word or number."* At every insertion depth, the gather and scatter steps run over every prompt position, including the visual tokens at the positions where the backbone's processor places them. The sequence read by the adapters therefore holds the video, the question, and the instruction, and the ground-truth answer stays outside the sequence at training time as at inference. The codes at each depth are thus a function of the prompt, and the answer is produced afterwards through the frozen `lm_head`. All arms decode greedily, and a prediction counts as correct if its first word matches the ground truth after lowercasing and removing punctuation. Trained arms produce their answer through the backbone's frozen `lm_head`. Every arm decodes at most three new tokens, which suffices under the first-word rule. Training and evaluation are both restricted to the open-ended descriptive questions; the explanatory, predictive, and counterfactual splits of CLEVRER are not used. AFTERIMAGE (7 insertion depths, 0.25% of backbone parameters) and the LoRA baseline (rank 32 on the attention projections of all blocks, 0.24%) are trained on the same data for the same number of epochs to convergence. For every trained arm, the reported checkpoint is the one with the best validation score, following the checkpoint rule of Appendix D. Both arms use AdamW at a constant learning rate of 1×10^{-4}, no scheduler, no weight decay, and batch size 8 without gradient accumulation. Both arms are trained on two H100 GPUs. Each trainable arm (LoRA, AFTERIMAGE, and the continuous bottleneck) is trained three times with different seeds, and the reported accuracies are the mean$\pm$std over the three runs. The frozen backbone is evaluated once.

E DATASET DETAILS

This appendix describes the brain MRI corpus, its subject-level splits, the diagnostic labels and their encoding, and preprocessing. All statistics in this section are computed from the manifests loaded by the training run rather than from the published descriptions of the source studies. We pool six public studies (ABCD, ADNI, AIBL, PPMI, HCP-Aging, HCP-Development; n=39,997 scans from 16,189 subjects, ages 5.6–104), since each study covers only part of the lifespan.

Splits. Each subject is assigned whole to one split. Every scan of a subject therefore travels with the subject, and the three splits share no subject. Training holds 14,943 subjects and 36,892 scans, validation 447 subjects and 1,202 scans, and test 799 subjects and 1,903 scans; together the splits keep every scan of the corpus. Within validation and test, a balanced *core* subset is marked for per-condition reporting. The core subset takes two scans per (age, sex, tag) cell in validation and three in test where the corpus allows, and takes sparse cells in full, giving 813 and 1,362 scans. The full sets are used where only held-out data is needed. Cells are filled preferentially with subjects who have few scans, and a held-out subject therefore carries fewer scans on average than a training subject. Relative to training, the held-out splits are enriched for the impaired tags, which lets condition

adherence be measured beyond the all-zero (CN) encoding. Table 8 gives the scan count of every diagnostic tag in each split.

Table 8: Scans per diagnostic tag in each split. Core is the balanced subset used for per-condition reporting.

Tag	Train	Val (full)	Test (full)	Test (core)
CN	29,126	309	481	467
EMCI	2,498	279	443	221
MCI	2,941	151	260	234
LMCI	933	267	437	215
AD	1,394	196	282	225
Total	36,892	1,202	1,903	1,362

E.1 Diagnostic labels

Table 9 inventories the raw `research_group` values of every source study and marks the four that receive a nonzero indicator.

Table 9: `research_group` values as they appear in the manifests. Only the four boldface Alzheimer's-spectrum categories are encoded as nonzero indicators; every other value, and every scan from a study with no label field, receives the all-zero (control) indicator.

Study	`research_group` inventory
ADNI	CN 4909, **MCI** 3352, **EMCI** 3220, **AD** 1872, **LMCI** 1637, SMC 525, Patient 9
PPMI	PD 542, Control 135, Prodromal 82, SWEDD 76
ABCD, AIBL, HCP-Aging, HCP-Development	no label field

Condition encoding. Each scan yields a six-dimensional condition vector: age in years, binary sex, and a four-dimensional indicator built as $\mathbf{1}[\texttt{research_group} = t]$ for $t \in (\text{EMCI}, \text{MCI}, \text{LMCI}, \text{AD})$.

The all-zero (control) code. The all-zero vector is assigned to every scan not tagged EMCI, MCI, LMCI, or AD. The assignment covers ADNI CN and SMC, PPMI Control, PD, Prodromal, and SWEDD, and every scan from a study without a diagnostic label field. Control in our setting therefore denotes the absence of an Alzheimer's-spectrum label rather than verified health, since cohorts labelled under other schemes enter the control group. The prompt phrase `no Alzheimer's-spectrum diagnosis` is the surface form of the all-zero tag (Appendix G). The impaired columns of Table 1 are restricted to the 70–99 age band, the band occupied by the impaired conditions.

E.2 Preprocessing

Volumes are RAS-oriented, resampled to $1\,\text{mm}^3$ isotropic spacing, cropped to $160 \times 192 \times 176$, and intensity-normalized to $[0, 1]$ by percentile scaling (0th–99.5th).

F Metric Details

This appendix specifies the three brain MRI metrics, each computed by an evaluator trained separately from the generator. The evaluators operate in the latent space of the same autoencoder in which the diffusion model generates, and each task has its own evaluator checkpoint. Every generated volume is paired with its requested condition, and each metric compares the evaluator's reading of the volume with the requested condition.

Evaluator training. The three evaluators share one design. The frozen autoencoder of the latent diffusion model encodes a volume into a 32-channel latent. A 3D convolutional encoder maps the latent to a 512-dimensional feature vector through three convolution blocks with group normalization and SiLU activations, two of them strided, followed by global average pooling and a linear projection. A single linear head sits on the features, and one model is trained per task. The age model regresses age with an L_1 loss, the sex model is trained with binary cross entropy, and the model supplying FID features classifies the five diagnostic groups with cross entropy. The FID feature space is therefore tuned to brain MRI rather than to natural images. Training uses real volumes from the same corpus and the same preprocessing as diffusion training, for 300 epochs with batch size 8, learning rate 10^{-5}, no weight decay, and seed 42; the checkpoint with the best validation score is kept. The Real row of Table 1 scores real scans drawn from the validation split. On the validation scans, the evaluators reach 7.04 years of age error and 0.86 sex accuracy, which gives a reference level for generated volumes on the two measures.

Fréchet distance in the evaluator feature space. FID is one Fréchet distance per reported cell of Table 1 (model $\times$ cohort $\times$ Syn or CF column), computed in the 512-dimensional evaluator feature space. All generated volumes in a cell are pooled and compared with an equal-sized sample of real volumes from the training split, drawn with the sampling seed fixed at 42. For counterfactual cells, the generated set consists of the edits of the real scans in the test core of Appendix E (1,362 scans), again pooled rather than split by age, sex, or tag within the cell. The Real row scores validation scans against the same training reference and is therefore not a floor on the FID column. Age MAE and sex accuracy remain means over the individual volumes in each cell and are not Fréchet estimates. The feature space is the diagnosis classifier above rather than an Inception network. The FID values therefore index relative fidelity within Table 1 and are not comparable with published FID. A Fréchet estimate over roughly a thousand samples in 512 dimensions is also biased upward, and the bias affects every arm of a cell similarly.

Age mean absolute error. The latent age regressor predicts the age of each generated volume, and the error is the absolute difference from the requested age. A volume requested at age 70 and read as 66.12, for example, contributes 3.88 years. The reported value is the mean over the volumes in the set.

Sex accuracy. The latent sex classifier predicts the sex of each generated volume, and accuracy is the fraction matching the requested sex.

Sample size. Counterfactual columns in Table 1 use one edit per real scan in the test core (1,362 scans; three per (age, sex, tag) cell in Appendix E). CF-A, CF-S, and CF-C therefore each contribute 1,362 volumes before the CN / impaired split of the table. Synthesis columns use the generated volumes for the same reporting conditions. FID uses the full pooled set in each table cell against a training-split reference of equal size; age MAE and sex accuracy average over the same volumes.

G Prompt Templates

This appendix gives the prompt templates through which the condition vector of Appendix E reaches the backbone. The condition vector is rendered into a single sentence before the forward pass. The rendering is not unique. We write five sentence templates and, for each diagnostic tag, five phrasings of the tag, and each rendering draws one template and one phrasing uniformly at random. A condition vector therefore maps to a set of equivalent strings rather than to one string, which discourages the conditioning signal from depending on a single surface form. For the code analysis of Appendix H, the 25 resulting combinations are split into ten synthesizing the samples on which signatures are selected and fifteen synthesizing the samples on which signatures are scored. Sex is 0 for male and 1 for female. The all-zero diagnostic tag is phrased in Alzheimer's-spectrum terms, for example `no Alzheimer's-spectrum diagnosis`; the tag is the control class of Appendix E and denotes the absence of an Alzheimer's-spectrum label rather than verified health. The table below gives one template and one phrasing per tag; the remaining four of each vary the wording while holding the expressed age, sex, and diagnosis fixed.

Template (1 of 5)	`Provide a brief description in formal neuroradiological detail of the findings observed on a 3D brain MRI scan of a {age}-year-old {sex} presenting with {condition}.`
`{sex}`	`male` | `female`
`{condition}` (1 of 5 per tag)	`no Alzheimer's-spectrum diagnosis` (all-zero tag) `early mild cognitive impairment` (EMCI) `mild cognitive impairment` (MCI) `late mild cognitive impairment` (LMCI) `Alzheimer's disease` (AD)

For example, $[70, 0, 0, 0, 0, 0]$ renders as `...of a 70-year-old male presenting with no Alzheimer's-spectrum diagnosis.` under the template and phrasings shown, and $[90, 0, 0, 0, 0, 1]$ as `...of a 90-year-old male presenting with Alzheimer's disease.` The sentence is wrapped in the backbone's chat template before the forward pass, and the adapters read the residual stream of the wrapped sequence.

H TRIGRAM SIGNATURE EXTRACTION

This appendix specifies the signature-extraction procedure behind Figure 3, including the specificity formula, candidate thresholds, fallback rules, joint-presence threshold, two-motif selection, age-matching, and deduplication omitted from Section 2.4.

Input. Signature selection and evaluation draw on disjoint surface forms of the same conditions. The condition vector is rendered through one of 25 combinations of five sentence templates and five per-tag phrasings (Appendix G). Ten combinations supply the samples on which a signature is selected, and the remaining fifteen supply the samples on which the signature is scored. The trained model is run over the evaluation conditions with code export enabled. The run produces one record per condition, holding the codebook IDs at each of the $|\mathcal{S}| = 8$ inserted adapters, $\kappa^{(i)} = (k_1^{(i)}, \ldots, k_{16}^{(i)})$, together with the age, sex, and diagnostic tag of the condition.

Trigrams. A trigram is a consecutive three-code run within the code sequence of a single depth, tagged with the depth at which the run was observed:

$$\tau = \big(i,\ k_j^{(i)},\ k_{j+1}^{(i)},\ k_{j+2}^{(i)}\big), \qquad j = 1, \ldots, K-2. \tag{8}$$

Each condition therefore contributes at most $|\mathcal{S}|(K-2) = 8 \times 14 = 112$ trigrams. We treat the trigrams of a condition as a set, discarding multiplicity and keeping presence only. Depth is part of the trigram identity, and the same three codes at two depths are two different trigrams.

Selecting a signature. Write $\mathrm{pres}(\tau, g)$ for the fraction of conditions in group g whose trigram set contains τ, and $\mathrm{joint}(\{\tau_1, \tau_2\}, g)$ for the fraction containing both. For each group g:

1. **Candidates.** $C_g = \{\tau : \mathrm{pres}(\tau, g) \geq 0.40\}$. If empty, fall back to the eight most frequent trigrams in g.
2. **First trigram.** $\tau_1 = \arg\max_{\tau \in C_g} \big[\mathrm{pres}(\tau, g) - \max_{g' \neq g} \mathrm{pres}(\tau, g')\big]$.
3. **Second trigram.** Among $\tau \in C_g \setminus \{\tau_1\}$ with $\mathrm{joint}(\{\tau_1, \tau\}, g) \geq 0.30$, take the τ_2 maximizing $\mathrm{joint}(\{\tau_1, \tau\}, g) - \max_{g' \neq g} \mathrm{joint}(\{\tau_1, \tau\}, g')$. Overlap is permitted; τ_1 and τ_2 may share codes or chain into a four- or five-code motif. If no candidate clears the threshold, the signature falls back to τ_1 alone. The 0.40 candidate threshold and the 0.30 joint-presence floor are the two free parameters of the procedure. Both are inert for the diagnostic tags, which select the same pair for every group across candidate thresholds from 0.30 to 0.50 and joint floors from 0.20 to 0.40. Within the same range, the 20s, 40s, and 50s decade signatures change, and the 40s and 50s decades also fail the permutation null below.

The reported specificity is $\mathrm{joint}(\{\tau_1, \tau_2\}, g) - \max_{g' \neq g} \mathrm{joint}(\{\tau_1, \tau_2\}, g')$, and the heatmap cell at row g, column g' is $100 \cdot \mathrm{joint}(\{\tau_1, \tau_2\}, g')$. Selection maximizes the margin between a group and its nearest comparison group on the ten selection templates, whereas every value reported in Figure 3 is measured on the fifteen evaluation templates withheld from selection. The off-diagonal leakage is

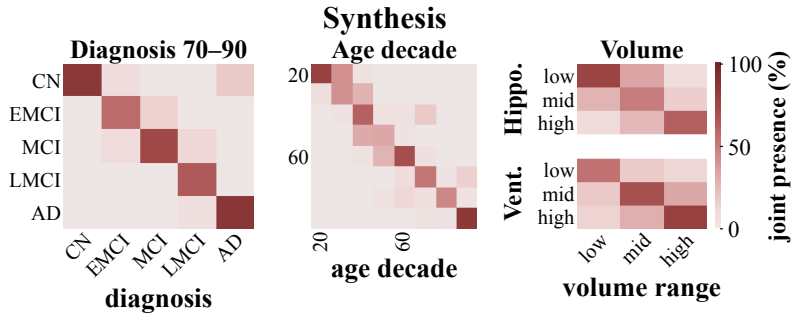


Figure 8: **MRI code signatures on the 10 selection templates (synthesis).** Same Synthesis panels as Figure 3, scored on the 10 selection templates rather than the 15 held-out evaluation templates. The block-diagonal structure resembles that of the held-out figure.

therefore a held-out quantity rather than an optimized one. Figure 8 shows the Synthesis panels on the 10 selection templates. The pattern resembles the held-out one, which is consistent with motifs stable across surface forms.

Permutation null. A label-permutation null tests whether the observed specificity exceeds what the selector recovers from shuffled labels. Selection uses group labels, and high specificity alone therefore does not rule out group structure invented by the selector from noise. For each group in each panel of Figure 3 (and in the sex and morphometric appendix panels), we shuffle condition labels within the panel while preserving group sizes, rerun the two-trigram selection above, and record the resulting specificity, over $N{=}5000$ permutations. Figure 9 plots, for every group, the observed specificity against the 95th percentile of the null. Benjamini–Hochberg FDR q-values are computed across the 30 group-level tests, which cover the specificity of each group and not the individual off-diagonal cells of the heatmaps. All five diagnostic tags clear $q{<}0.05$, as do both sexes, most age decades, and most volume tertiles, whereas the 40s and 50s decades and the lowest ILV tertile do not. For most groups, the observed specificity therefore sits above what label-shuffled selection recovers, which indicates that the main heatmap associations are not an artifact of the selector alone; the mid-life decades remain close to the null.

Grouping.

- **Tags.** Restricted to ages $\{70, 75, 80, 85, 90\}$ and deduplicated to one condition per (age, sex, tag) cell, over CN, EMCI, MCI, LMCI, and AD. Age-matching makes the tag contrast interpretable. Without age-matching, a diagnostic signature could be an age signature, since impaired conditions occupy only the upper age range.
- **Decades.** CN conditions only, bucketed by $\lfloor \mathrm{age}/10 \rfloor$. Holding the tag fixed keeps the diagnosis axis from leaking into the age signature.
- **Sex.** CN only. Both sexes span the full age range, so the groups are age-balanced by construction.
- **Morphometric ranges.** For the hippocampus, amygdala, entorhinal cortex, and inferior lateral ventricles, each condition takes the mean FastSurfer volume of its generated samples, and the evaluation conditions are split into low, mid, and high thirds of the volume, separately per feature. Signatures are selected among the three ranges of one feature at a time.

Feature-neutral substitution and generated volume. Feature-neutral substitution measures how generated volume changes when the age and volume motifs are removed from the code sequence. Figure 10 reports the full hippocampus and ventricle-total grid, crossing the age-decade signature with the high / mid / low volume motifs. Each panel overwrites the two motifs with a feature-neutral motif and groups conditions by which motif is left in place. The grid therefore covers feature-neutral substitution alone and not the morphometric replacement of Figure 4. Across decades, the drift in generated volume changes sign, rising above the real band for the 20s and falling below the band for the 80s. The sign change is consistent with motifs associated with a decade-specific volume level rather than with a uniform shift in volume. In the 20s, the age motif alone already reaches the real level, as expected for the decade with the smallest ventricular enlargement. The dashed line and

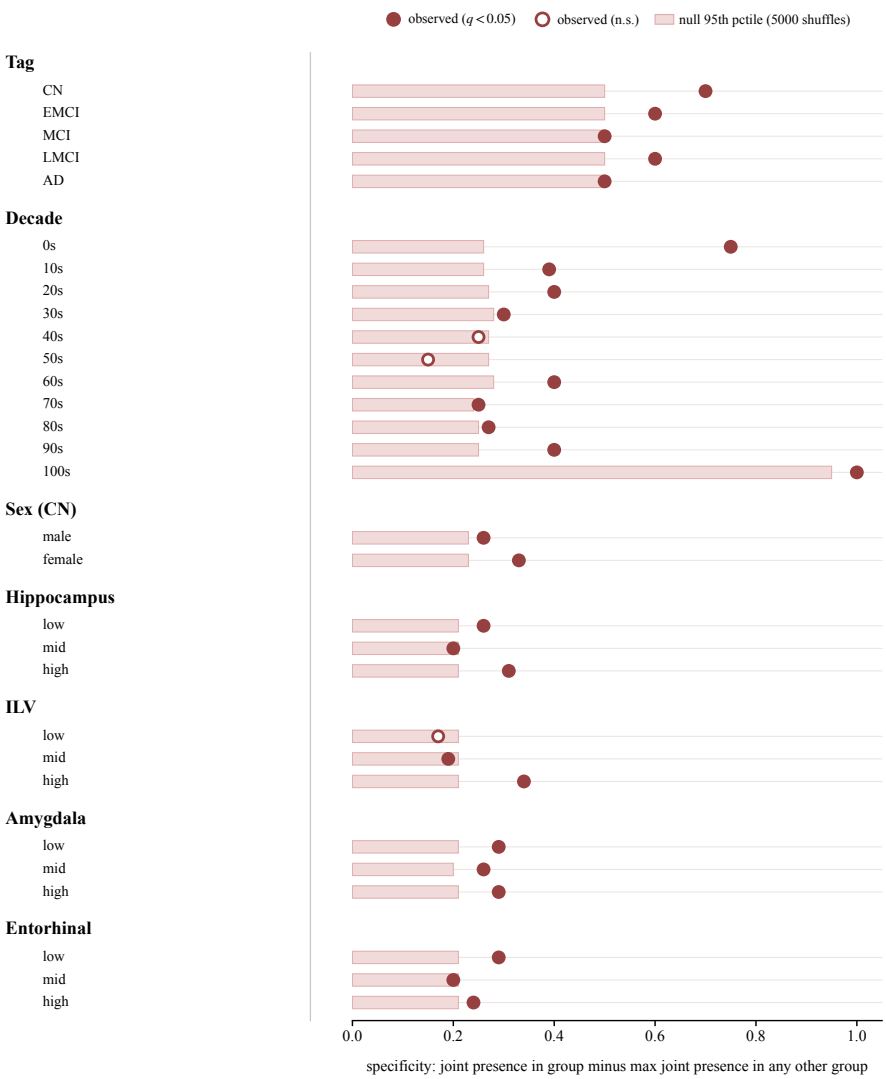


Figure 9: **Label-permutation null for MRI code-signature specificity.** For each group, the horizontal bar is the 95th percentile of specificity under 5000 within-panel label shuffles that rerun two-trigram selection; the marker is the observed specificity (filled if FDR $q<0.05$). All diagnostic-tag and sex signatures clear their null under FDR control, as do the age and morphometric signatures apart from the 40s and 50s decades and the lowest ILV tertile, giving 27 of 30 group-level tests at $q < 0.05$.

band mark the real FastSurfer median and IQR for the decade; the y-scale is shared within each morphometric row.

Motif substitution targets. A motif substitution replaces both trigrams of a signature pair, and the replacement is likewise a full pair. The left two panels of Figure 4 overwrite the currently firing source motif with a named *target* motif, namely the 20s or 80s age trigrams for ventricles and the CN or AD diagnosis trigrams for hippocampus. The forward pass then proceeds as usual, and codes at later depths are assigned from the modified stream. FastSurfer (Henschel et al., 2020) then measures the regional volumes of every generated scan, as for the intact model. The right panel of Figure 4 overwrites the age and high-ventricle motifs of the 80s conditions with a feature-neutral motif in the four combinations of which motif is left in place. A fifth condition keeps the age motif and replaces

Table 10: **Brain MRI motif substitution (Figure 4, right panel only).** Age MAE ↓ and sex accuracy ↑ for the last subplot of Figure 4: a *feature-neutral* replacement (codes not selected as any category signature) of the age and/or high-ventricle motifs on the 80s conditions, or the *age+low* replacement that keeps the age motif and overwrites the high-ventricle motif with the low-ventricle one. The left and middle panels of Figure 4 report morphometric volume only and are not included here. Intact CN and Impaired synthesis from Table 1 are shown for reference; all rows report mean±std over three runs. FID is not reported here, since motif overwrites change the target tags, so Fréchet distance against the intact reference is not a fair comparison.

Condition	Age MAE (y) ↓	Sex acc. (%) ↑
AFTERIMAGE (intact CN Syn)	11.8 ± 0.7	76.2 ± 0.8
AFTERIMAGE (intact Impaired Syn)	12.8 ± 1.0	84.0 ± 1.2
Feature-neutral motif swap	23.4 ± 4.7	65.2 ± 3.8
Age+low motif swap	19.3 ± 3.6	61.8 ± 2.9

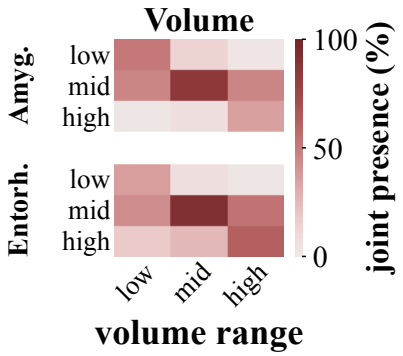


Figure 10: **Feature-neutral substitution and generated regional volume.** Rows: hippocampus and ventricle total. Columns: 20s / 60s / 80s age signatures, each paired with the matching high / mid / low volume motifs. Violins group conditions by which motifs are left in place (neither, age only, morpho only, both); markers show the median and IQR. Dashed line and band: real subjects in the same decade.

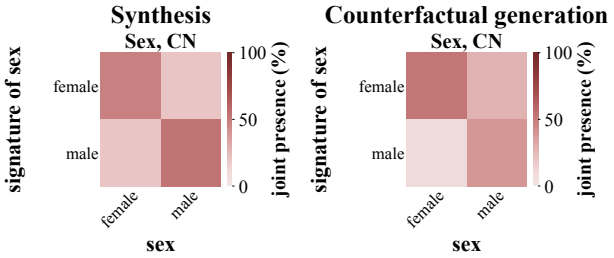
the high-ventricle motif with the low-ventricle one, and the panel reports generated ventricle volume against the real 80s reference. Table 10 collects age MAE and sex accuracy for the right panel of Figure 4 only, that is, for the feature-neutral and age+low motif substitutions on the 80s ventricle conditions. FID is omitted, since motif overwrites shift the target tags and a Fréchet distance against the intact reference is therefore not comparable. On robot action planning, the overwrite applies at every decision at which the task signature fires, throughout the episode. The replacement is either a *feature-neutral* motif (codes not selected as a signature for any category) or the signature of a task from a different LIBERO suite, and Figure 7 and Table 11 (c) report both.

Two further panels apply the same selection to other grouping variables. Figure 11a applies the selection to regional volumes, putting amygdala and entorhinal-cortex tertiles in place of the age decades. Figure 11b repeats the selection with sex as the grouping variable, holding the age range fixed across the two groups.

Reading the decade panel. Decade specificity is bimodal, unlike tag specificity. The extreme decades score highest, whereas the mid-life decades sit at $+0.15$ to $+0.30$. Adjacent decades share structure, as expected for an ordered variable, and a decade signature leaks more into neighbouring decades than into distant ones.



(a) **Amygdala / entorhinal volume.** Two within-depth trigrams per regional-volume tertile; rows are the tertile selected on, columns the evaluation tertile, cells the fraction of conditions containing both. Signatures are most frequent in their own range and leak mainly to adjacent ranges. Volumes are measured on generated samples.



(b) **Sex among CN conditions.** Two-trigram signatures are selected separately for each sex among CN conditions. Both groups span the full age range, which makes the comparison age-balanced by construction. Each sex carries its own signature, present in its own group and largely absent from the other.

Figure 11: **Code-signature panels.** Left: amygdala and entorhinal-cortex volume tertiles. Right: sex among CN conditions.

I From Commanded Actions to Human Phases

This appendix specifies how commanded actions are mapped to the human phases used by the action-phase signatures. Manipulation policies emit low-level end-effector commands, and reading the commands as human phases requires a map onto a finite phase vocabulary. Prior work builds the map with proprioceptive rules or with a VLM (Chen et al., 2025; Li et al., 2026). We follow the rule-based approach and specify a portable pipeline below (Algorithm 2).

Unit. The labeling unit is one *action chunk*, the sequence of commanded actions a policy predicts at a single decision (or, equivalently, a contiguous window of executed commands). Each chunk receives exactly one phase, and consecutive decisions may share a label when the behaviour continues.

Input. Write each command as $a_t = (\Delta\mathbf{x}_t,\ \Delta\boldsymbol{\omega}_t,\ g_t)$, with planar/vertical translation $\Delta\mathbf{x}_t \in \mathbb{R}^3$, rotation $\Delta\boldsymbol{\omega}_t \in \mathbb{R}^3$, and gripper scalar g_t (sign convention: positive = closing / closed). Any comparable proprioceptive stream can serve as input; joint velocities may replace $\Delta\mathbf{x}, \Delta\boldsymbol{\omega}$ as in Chen et al. (2025).

Features. From chunk $A = (a_t)_{t=1}^{N}$:

1. *Debounce* the binary gripper state $c_t = \mathbf{1}[g_t > 0]$ by dropping runs shorter than a width δ, which prevents chatter from counting as a transition.
2. Record grip events, where a close is $c\colon 0{\to}1$ and an open is $c\colon 1{\to}0$.
3. Set $\mathrm{holding}$ if the gripper is closed for more than half the chunk.
4. Aggregate motion: mean translation speed s, mean rotation rate r, net vertical $\Delta Z = \sum_t \Delta z_t$, and net planar travel $\Delta XY = \|(\sum_t \Delta x_t, \sum_t \Delta y_t)\|_2$.
5. Threshold with dataset statistics: $\mathrm{move} \Leftrightarrow s \geq \tau_{\mathrm{spd}}$, $\mathrm{rot} \Leftrightarrow r \geq \tau_{\mathrm{rot}}$, with $(\tau_{\mathrm{spd}}, \tau_{\mathrm{rot}})$ the medians (or other quantiles) of s and r over the corpus; set $\mathrm{vert} \Leftrightarrow |\Delta Z| > \Delta XY$.

Assignment. A fixed priority tree makes the labels mutually exclusive. The tree tests grip events first (close-only / open-only / both); otherwise, if holding, it branches on vertical vs. planar vs. rotational vs. still motion; otherwise it applies the same motion tests with an empty hand; and otherwise it returns idle. The concrete names in $\mathcal{P}$ (e.g. grasp, lift, transport, reach) may be swapped for another primitive set (Li et al., 2026) as long as the tree stays mutually exclusive.

Algorithm 2 Human phase from an action chunk. Gripper state and end-effector motion select one exclusive phase $p \in \mathcal{P}$, following rule-based interaction-phase segmentation (Chen et al., 2025) and reusable motion primitives (Li et al., 2026).

Require: chunk $A = (a_t)_{t=1}^{N}$ with $a_t = (\Delta\mathbf{x}_t, \Delta\boldsymbol{\omega}_t, g_t)$; thresholds $\tau_{\mathrm{spd}}, \tau_{\mathrm{rot}}$; debounce δ; vocabulary $\mathcal{P}$
Ensure: phase $p \in \mathcal{P}$

$c_t \leftarrow \mathbf{1}[g_t > 0]$; drop runs of length $<\delta$ ▷ debounce
close $\leftarrow \exists\, t{:}\,(c_{t-1}, c_t) = (0,1)$; open $\leftarrow \exists\, t{:}\,(c_{t-1}, c_t) = (1,0)$
holding $\leftarrow \frac{1}{N}\sum_t c_t > \frac{1}{2}$
$s \leftarrow \frac{1}{N}\sum_t \|\Delta\mathbf{x}_t\|_2$; $r \leftarrow \frac{1}{N}\sum_t \|\Delta\boldsymbol{\omega}_t\|_2$
$\Delta Z \leftarrow \sum_t \Delta z_t$; $\Delta XY \leftarrow \|(\sum_t \Delta x_t, \sum_t \Delta y_t)\|_2$
move $\leftarrow [s \geq \tau_{\mathrm{spd}}]$; rot $\leftarrow [r \geq \tau_{\mathrm{rot}}]$; vert $\leftarrow [|\Delta Z| > \Delta XY]$
if close $\wedge\ \neg$open **then**
    return grip-close phase ▷ e.g. grasp
else if open $\wedge\ \neg$close **then**
    return grip-open phase ▷ e.g. release
else if close $\wedge$ open **then**
    return both-transitions phase ▷ e.g. regrasp
else if holding **then**
    if vert **then**
        return held vertical phase by $\mathrm{sign}(\Delta Z)$ ▷ e.g. lift / place
    else if move **then**
        return held planar phase ▷ e.g. transport
    else if rot **then**
        return held rotational phase ▷ e.g. reorient
    else
        return held still phase ▷ e.g. hold
    end if
else if vert **then**
    return free vertical phase by $\mathrm{sign}(\Delta Z)$ ▷ e.g. ascend / descend
else if move **then**
    return free planar phase ▷ e.g. reach
else
    return idle phase
end if

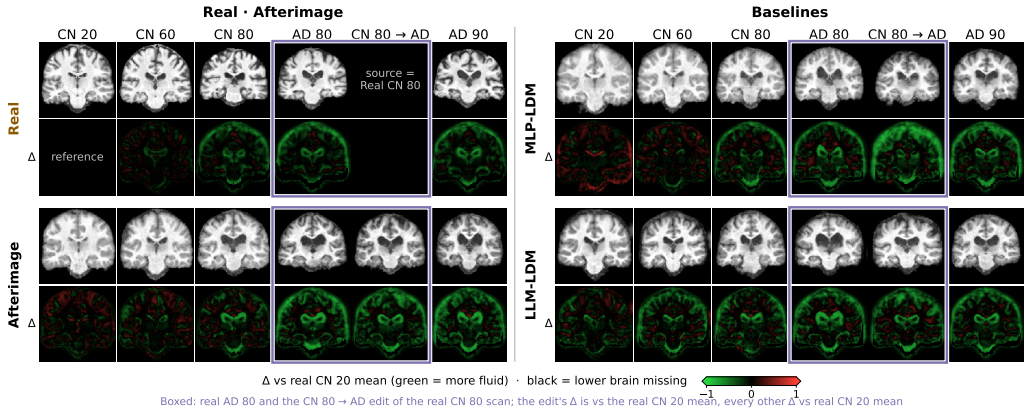
Reuse. Reapplying the pipeline to another robot or corpus takes four steps: (i) choose the chunking unit to match the controller, (ii) calibrate $\tau_{\mathrm{spd}}, \tau_{\mathrm{rot}}, \delta$ on the target corpus, (iii) keep grip events above motion so contact boundaries stay stable, (iv) optionally merge rare leaves into coarser bins if coverage is uneven.

J Additional Brain MRI Results

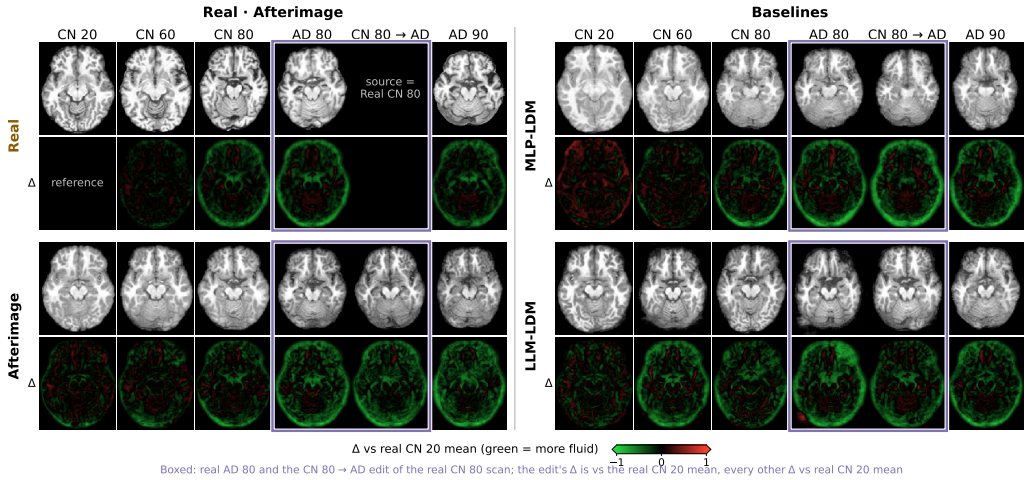
This appendix documents how the qualitative MRI samples are selected and compares regional volume trends between real and generated scans.

Qualitative sample selection. In Figure 2, real difference maps average eight subjects per condition, whereas model difference maps use the displayed sample. In the difference maps, green marks darker voxels (consistent with more fluid) and red marks brighter voxels. Voxels where a scan lacks the lower brain are shown in black, and the cyan outline marks the typical hippocampus location of a healthy young adult. The counterfactual column edits the real CN 80 scan (ADNI, age 81, sex 0) toward AD. The latent of the scan is noised to step 600 of 1000 and denoised under the AD condition. Counterfactual difference maps are computed against the mean of eight real CN 20 scans, like all other difference maps. Figure 12 shows the same scans in full coronal, axial, and sagittal slices.

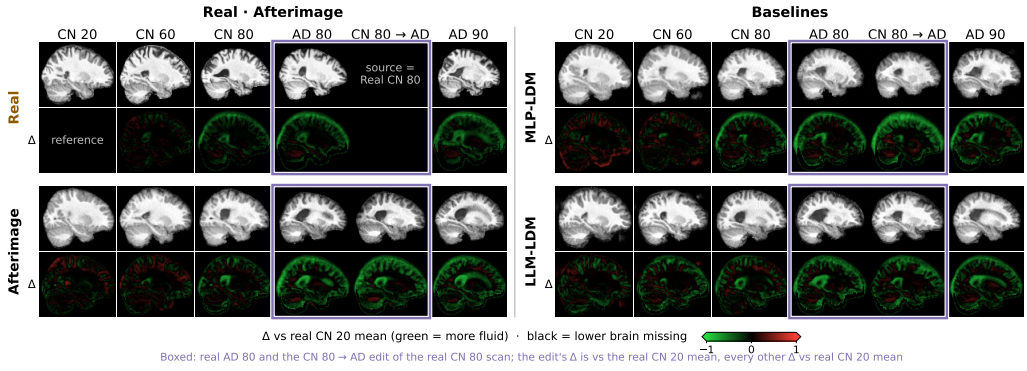
Morphometric trends. The morphometric comparison tests whether generated scans reproduce the regional volume trends of real scans across age and diagnosis. We segment real scans and gen-



(a) Coronal slice ($y = 88$).



(b) Axial slice ($z = 68$).



(c) Sagittal slice ($x = 105$).

Figure 12: **Multi-view qualitative MRI synthesis and condition editing.** Coronal, axial, and sagittal views of the same scans as the main text, each with a difference map against the mean of eight real CN 20 scans. Green marks lower normalized intensity, consistent with more CSF-like fluid, and red marks higher intensity. Black areas are outside the available lower-brain field of view, not generated anatomy.

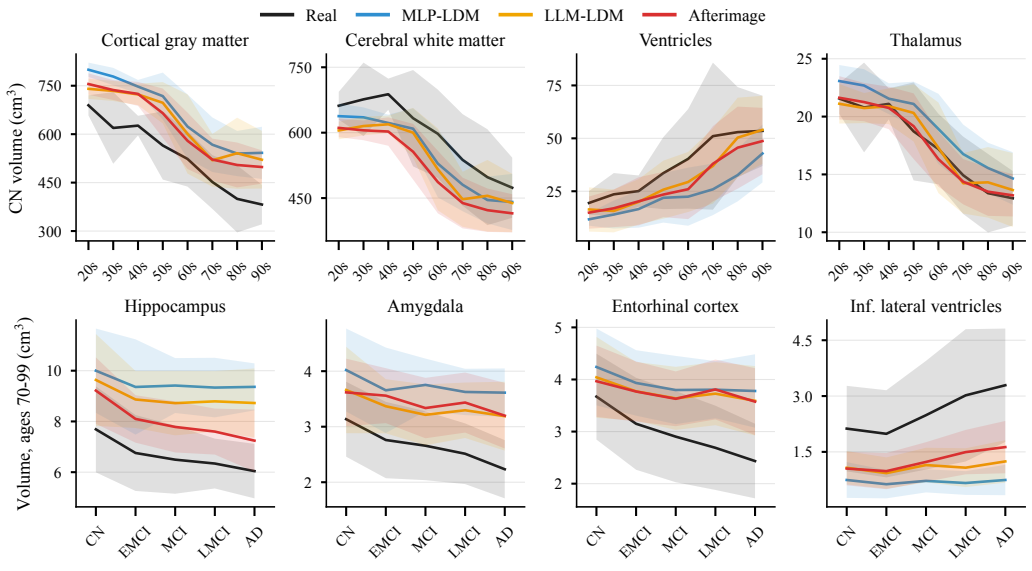


Figure 13: **Regional morphometric trends in real and generated brain MRI.** FastSurfer-derived regional volumes across age decades for CN conditions and across diagnostic groups at ages 70–99. Among the generators, AFTERIMAGE follows the broad age-associated trends most closely and shows the clearest hippocampal CN-to-AD decline and inferior-lateral-ventricle enlargement. Amygdala and entorhinal changes remain underrepresented relative to real scans.

erated samples with FastSurfer and compare regional volumes (Figure 13). Real scans pass through the same preprocessing and intensity transform as generated volumes before segmentation, which makes the two sets of volumes directly comparable. Structures changing mainly with age are plotted against age decade for CN conditions. Medial temporal structures separating diagnostic groups are plotted against diagnosis at ages 70 to 99, the range in which every model generates impaired cohorts. All three models reproduce the direction of the age trend in every plotted structure. Cortical gray matter, white matter, and thalamus shrink, and the ventricles enlarge, across the lifespan. The generated thalamic trajectory is close to the real one, and AFTERIMAGE follows the overall real decline most closely. Generated ventricles enlarge at a similar or faster rate than real ventricles while remaining smaller at nearly every age. The generated cortex is larger than the real cortex at every age, and its decline is shallower. Across diagnoses, real scans show a steady CN-to-AD decrease in hippocampus, amygdala, and entorhinal cortex volume, together with enlarging inferior lateral ventricles. AFTERIMAGE recovers a clear hippocampal decline and temporal-horn enlargement across diagnostic groups, whereas the amygdala and entorhinal cortex remain flatter than in real scans. All models still overestimate absolute medial temporal volumes. Real CN groups follow the all-zero conditioning label and therefore include some subjects whose source study records a diagnosis outside the Alzheimer's spectrum.

K ADDITIONAL ROBOTICS RESULTS

This appendix reports the complete robot results and the code-signature analyses behind the main-text robot figures. Table 11 reports every LIBERO-Pro suite, including Position OOD and the Non-causal ablation for both backbones, and motif substitution for $\pi_{0.5}$ + AFTERIMAGE under Language (panel c). Table 2 in the main text shows all four suites for IID, Language, and Task. The Non-causal arm trains the same AFTERIMAGE adapter without the causal mask on the gather refiner. Suite means of the Non-causal arm also appear in the appendix ablation bars (Figures 29 and 30).

Task signatures in the code stream. Task signatures measure how well the last-depth code sequence separates the forty LIBERO tasks. Figures 14a and 14b read the assigned codes back as per-task signatures, one figure per backbone. A *trigram* is a run of three consecutive codes within the slot sequence at one insertion depth, as in Appendix H. The robot analysis uses the last insertion depth, whose codes vary across decisions in both backbones. A *decision* is one inference. The

Table 11: **Full robotics results, in-distribution and out-of-distribution.** Success rate (%) ↑ as mean±std over three runs. Panel (a) is standard LIBERO, panel (b) LIBERO-Pro, and panel (c) motif substitution under Language for $\pi_{0.5}$ + AFTERIMAGE. Each backbone is reported alone, with LoRA, Cont. (no quantizer), Cont. + noise, Non-causal (no causal gather mask), and AFTERIMAGE. LIBERO-10 is the long-horizon two-clause suite. Panel (c) reports Language success after overwriting the task signature with a feature-neutral or other-task motif; Mean is over the four suites (std over suites), matching Figure 7; Appendix K.4 reports the same-scene swap. Best mean per column within each backbone among reported arms in bold, ties unbolded.

(a) IID: LIBERO.

Arm	Spatial	Object	Goal	LIBERO-10
SmolVLA (Shukor et al., 2025)	83.2±0.9	91.5±0.5	86.1±1.1	52.4±1.2
SmolVLA + LoRA	96.0±0.4	98.0±0.3	**98.5**±0.3	**91.8**±0.5
SmolVLA + Cont.	95.3±0.6	**98.4**±0.3	96.9±0.5	89.6±0.8
SmolVLA + Non-causal	87.4±0.7	95.2±0.4	91.6±0.6	87.8±0.9
SmolVLA + AFTERIMAGE	**96.5**±0.4	97.4±0.2	98.2±0.3	91.0±0.7
$\pi_{0.5}$ (Physical Intelligence, 2025)	98.2±0.5	98.5±0.3	98.3±0.6	97.4±0.8
$\pi_{0.5}$ + LoRA	95.9±0.5	97.4±0.3	99.0±0.2	97.7±0.4
$\pi_{0.5}$ + Cont.	95.7±0.5	96.1±0.4	97.4±0.3	98.3±0.4
$\pi_{0.5}$ + Cont. + noise	96.0±0.5	96.4±0.4	97.6±0.3	98.1±0.4
$\pi_{0.5}$ + Non-causal	95.5±0.5	97.0±0.3	98.8±0.4	97.4±0.6
$\pi_{0.5}$ + AFTERIMAGE	**98.6**±0.2	**99.3**±0.1	**99.4**±0.2	**98.8**±0.3

(b) OOD: LIBERO-Pro.

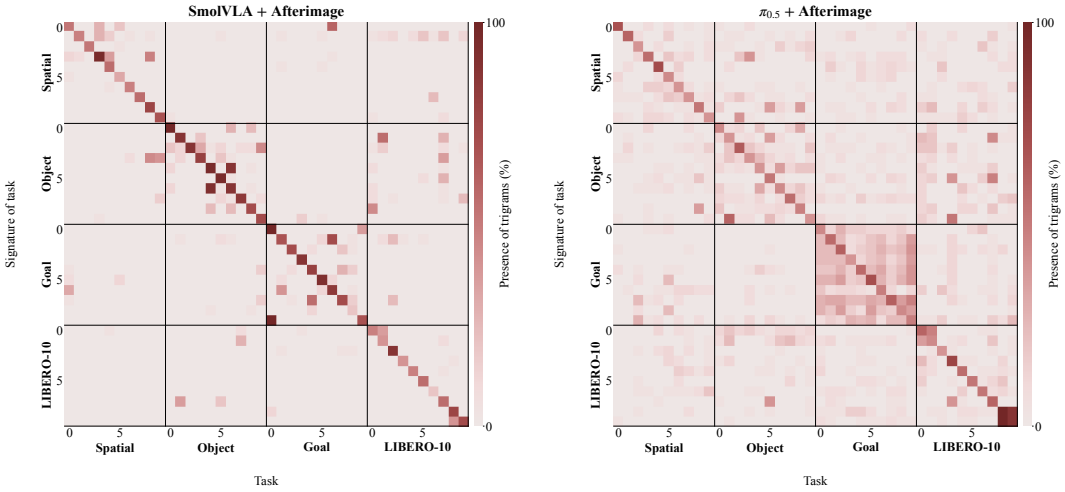
	Language				Position				Task			
Arm	Spatial	Object	Goal	LIBERO-10	Spatial	Object	Goal	LIBERO-10	Spatial	Object	Goal	LIBERO-10
SmolVLA	43.8±2.4	76.2±1.6	39.1±3.0	28.6±2.2	4.8±1.5	0.2±0.2	0.1±0.1	0.0±0.0	17.9±2.1	6.4±1.4	13.5±1.9	4.1±1.2
SmolVLA + LoRA	61.5±1.9	83.5±1.4	63.0±2.1	61.6±1.8	8.9±1.8	0.4±0.3	2.3±1.0	0.0±0.0	6.4±1.5	0.2±0.2	9.6±1.6	**8.5**±1.2
SmolVLA + Cont.	75.0±1.8	88.8±1.2	55.4±2.3	54.8±1.9	14.5±2.2	0.3±0.2	2.9±1.1	0.0±0.0	34.5±2.3	2.6±0.8	12.5±1.5	3.7±1.0
SmolVLA + Non-causal	62.6±1.9	81.6±1.3	62.4±2.1	61.4±1.8	10.2±2.0	0.4±0.3	0.4±0.3	0.0±0.0	5.0±1.4	0.0±0.0	9.8±1.6	7.0±1.2
SmolVLA + AFTERIMAGE	**93.8**±1.2	**96.2**±0.8	**69.4**±2.0	**72.6**±1.6	**25.2**±2.4	**1.4**±0.4	**7.2**±1.4	**0.6**±0.3	**49.6**±2.3	**9.2**±1.2	**18.8**±1.7	8.4±1.1
$\pi_{0.5}$	76.6±2.0	95.1±1.1	41.2±3.0	76.9±1.8	45.1±2.9	20.5±2.3	17.2±2.0	7.3±1.6	20.6±2.2	6.8±1.5	8.1±1.7	16.4±2.0
$\pi_{0.5}$ + LoRA	85.3±1.5	96.1±0.9	79.9±2.0	82.9±1.6	60.2±2.3	41.5±2.4	27.1±2.0	33.0±2.2	40.4±2.0	23.2±1.9	27.2±2.0	24.8±1.8
$\pi_{0.5}$ + Cont.	83.8±1.6	93.8±1.0	80.5±2.1	84.1±1.5	58.7±2.4	39.3±2.5	27.3±2.2	29.4±2.3	40.6±2.1	23.8±1.9	26.7±1.8	25.0±2.0
$\pi_{0.5}$ + Cont. + noise	89.4±1.6	89.8±1.0	81.0±2.1	88.7±1.5	60.1±2.4	38.8±2.5	28.0±2.2	30.1±2.3	41.0±2.1	24.1±1.9	26.2±1.8	25.4±2.0
$\pi_{0.5}$ + Non-causal	84.3±1.7	95.3±0.9	80.3±2.2	82.0±1.6	59.6±2.3	39.9±2.4	25.1±2.1	30.5±2.2	38.4±2.0	24.1±1.9	26.4±1.8	23.8±2.1
$\pi_{0.5}$ + AFTERIMAGE	**91.4**±1.3	**99.5**±0.4	**89.2**±1.8	**90.6**±1.2	**74.8**±2.0	**54.6**±2.8	**41.2**±2.3	**42.0**±2.4	**49.8**±1.9	**33.4**±2.2	**39.0**±2.0	**38.6**±1.8

(c) Motif substitution under Language ($\pi_{0.5}$ + AFTERIMAGE).

Overwrite	Spatial	Object	Goal	LIBERO-10	Mean
Intact AFTERIMAGE	91.4±1.3	99.5±0.4	89.2±1.8	90.6±1.2	92.7±4.6
Feature-neutral motif	43.8±2.8	33.9±2.6	39.3±2.9	29.5±2.4	36.6±6.2
Other-task motif	38.9±2.5	32.0±2.4	35.8±2.7	29.0±2.2	33.9±4.3

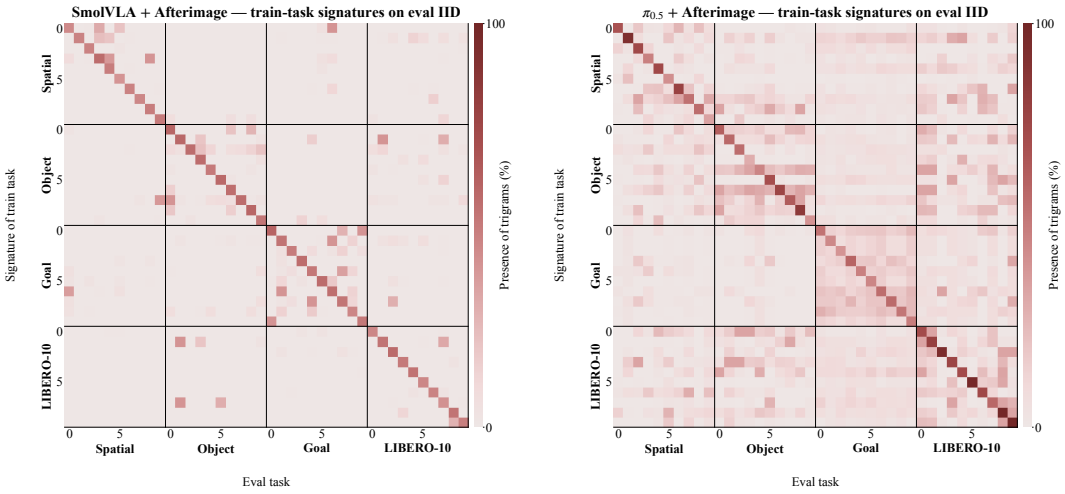
code vector is debounced, which makes consecutive steps sharing a code vector count once, and each decision contributes the six trigrams of its eight last-layer slots. The unit is the *episode*, whose trigram set is the union over its decisions; a trigram is present in an episode if the trigram appears in at least one decision of the episode. Every cell is a percentage whose denominator is the number of episodes behind its *column* rather than its row. For the task figures, the denominator is the episodes of the column task, half of which are held out where selection and reporting are split. For the object figures, the denominator is every episode of every task whose scene contains the column object; the denominator therefore grows with the number of scenes holding the object and differs across columns. Trigrams present in more than 5% of all episodes are set aside first, which prevents a motif common to the whole code stream from being selected as the signature of any task. For each task, the signature is the two remaining trigrams most often present in one half of its episodes, and the reported value is the percentage of the held-out half containing both trigrams. Both halves are training rollouts, and the validation episodes take no part in selection in any robot figure. The diagonal is the signature of a task on its own episodes, and every other cell is the leakage of the signature into another task. All forty tasks appear in one matrix per backbone, ordered by suite, and leakage within a suite and across suites are both visible. SmolVLA separates its tasks sharply, and its leakage concentrates within a suite rather than across suites.

Do the signatures survive evaluation? Figures 15a–17 take the signatures selected on the train-set rollouts and measure the signatures, unchanged, on the evaluation rollouts. Each cell is the per-



(a) SmolVLA + AFTERIMAGE. (b) $\pi_{0.5}$ + AFTERIMAGE.

Figure 14: **Held-out task-signature transfer.** Two trigrams are selected per task from training rollouts and evaluated unchanged on held-out episodes; rows denote the task used for selection, columns the evaluation task, and cells the fraction of episodes containing both trigrams. Diagonal dominance indicates task specificity, and off-diagonal values quantify leakage. Both backbones share the same construction and color scale.



(a) SmolVLA + AFTERIMAGE. (b) $\pi_{0.5}$ + AFTERIMAGE.

Figure 15: **Train-task signatures on eval IID.** Rows are the two most frequent last-layer trigrams of each train task, selected on train-set episodes; columns are the evaluation tasks; each cell is the percentage of the eval task's episodes containing both trigrams. Construction, exclusion rule, and color scale follow the per-task signature figures above.

centage of an eval task's episodes containing both trigrams of a train task's signature. For SmolVLA, a signature transfers to held-out episodes of the same task with little loss and remains well above its leakage into other tasks. Under the three LIBERO-Pro shifts, signature presence falls while its ratio to the off-diagonal holds; the SmolVLA signatures thus become less present without becoming less specific. The $\pi_{0.5}$ panels report the same measurement for the second backbone. The greyed block in the Position panel marks LIBERO-Spatial, where only one of its ten tasks has Position-shift evaluation data.

K.1 SIGNATURES FOR ENTITIES THAT SHARE THEIR UNITS

This subsection extends signature selection to entities whose units overlap, such as scene objects shared across tasks. Task signatures are read off a partition, in which every episode belongs to

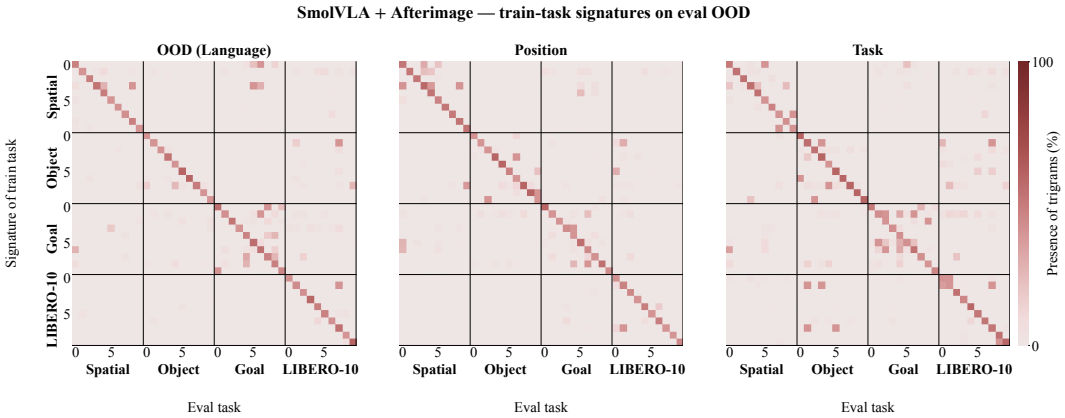


Figure 16: **Train-task signatures on eval OOD, SmolVLA + AFTERIMAGE.** Rows are the two most frequent last-layer trigrams of each train task, selected on training episodes and applied here unchanged; columns are the evaluation tasks under the three LIBERO-Pro shifts, which paraphrase the instruction, relocate the objects, and redesign the task logic. Each cell is the percentage of an eval task's episodes containing both trigrams, drawn on the color scale of the in-distribution figure above. The signatures remain concentrated on their own task under each shift, which indicates that the code signatures stay associated with the task across changes in wording, layout, and goal.

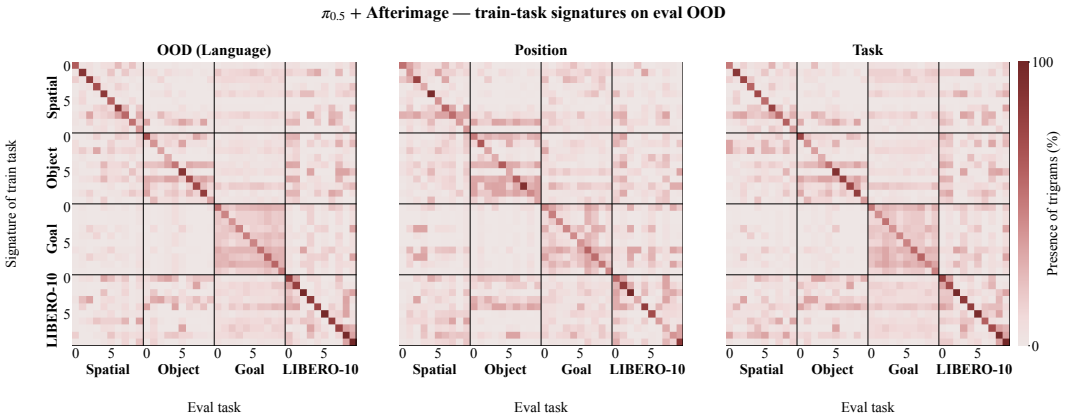
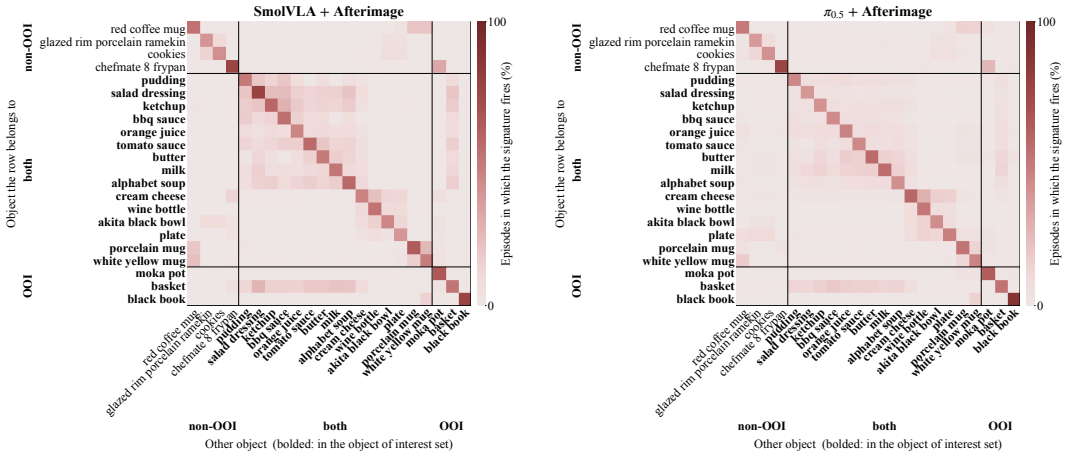


Figure 17: **Train-task signatures on eval OOD, $\pi_{0.5}$ + AFTERIMAGE.** The same train-selected trigrams, measured unchanged under the three LIBERO-Pro shifts of language, position, and task logic, and drawn as for the SmolVLA arm above. Each cell is the percentage of an eval task's episodes containing both trigrams, and the signatures remain most present on their own task under each shift.

exactly one task. A motif frequent in one task's episodes and rare elsewhere is then the signature of the task. The procedure below covers the general case, in which each entity occupies some subset of the units and the subsets may overlap freely. Telling co-occurring entities apart then requires a contrast against the remainder and an exclusivity constraint. The procedure is stated for arbitrary entities and units and instantiated here on the scene objects of the robot domain.

Units. A unit is the grouping over which presence is counted; in the robot domain, a unit is one episode. Within a unit, a step is kept only when the slot vector at the last insertion depth differs from the previous kept step. A unit therefore contributes one entry per inference rather than per environment step. Each kept vector of K slots yields $K - 2$ consecutive trigrams, and the trigram set of the unit is the union of the trigrams over its kept vectors.

Contrast. The contrast score favours a trigram frequent on units containing an entity and rare on units without the entity. For an entity C, let W_C be the units containing C and O_C the remaining units, and for a trigram g let $\mathrm{cov}(g, C)$ and $\mathrm{pol}(g, C)$ be the percentages of W_C and of O_C whose



(a) SmolVLA + AFTERIMAGE. (b) $\pi_{0.5}$ + AFTERIMAGE.

Figure 18: **Object trigram signatures.** Each row is the pair of trigrams assigned exclusively to the row object, and each cell is the percentage of the column object's episodes in which both trigrams fire, taken over the 22 LIBERO scene objects across all four suites. Objects appearing across varied scenes separate clearly, and the remaining leakage falls on scene-mates rather than forming contiguous blocks.

set contains g, with $\mathrm{glob}(g)$ the percentage over all units. Candidates are scored by

$$\mathrm{lift}(g, C) \;=\; \frac{\mathrm{cov}(g, C)}{\max\big(\mathrm{pol}(g, C),\, \varepsilon\big)}, \qquad \varepsilon = 0.5, \tag{9}$$

keeping candidates with $\mathrm{cov}(g, C) \geq 5\%$, which keeps the motif frequent enough on the entity to be read, and $\mathrm{glob}(g) \leq 8\%$, which withholds motifs common to the whole code stream from every entity. The floor ε bounds the score of a trigram absent from O_C.

Exclusive assignment. Exclusive assignment separates co-occurring entities by giving each trigram to at most one entity. The surviving pairs (C, g) are ranked by descending lift, breaking ties by coverage and then by name, and the list is walked once. A trigram g goes to C if g is unclaimed and C holds fewer than $k = 2$ trigrams, and every trigram is therefore owned by at most one entity. The exclusivity constraint makes the selection informative when entities co-occur. The constraint places members of a co-occurring group on different motifs rather than on the shared motif they would otherwise all take. A unit fires for C when *both* assigned trigrams are present, following the joint-presence rule of Appendix H.

Object signatures in the code stream. Figures 18a and 18b apply the procedure to the 22 LIBERO scene objects over all four suites at once, using the $K = 8$ slots of the last insertion depth.

K.2 ACTION-PHASE SIGNATURES

Action-phase signatures apply the entity procedure to human phases derived from commanded actions. Figure 5 (Actions) applies the exclusive top-two trigram rule of Appendix K.1 to the phases produced by Algorithm 2. In the train block, a cell is the fraction of labeled chunks in which the row signature fires, whereas the IID and OOD blocks report signature presence and joint presence on evaluation episodes without phase labels. Figure 19 repeats the whole measurement for the Non-causal arm of both backbones.

Presence metric for the robot signatures. The robot and brain MRI signature figures use the same presence metric, scoring a signature as present when both of its trigrams appear, following the joint-presence rule of Appendix H. The color scale spans the full 0–100% range in both cases, and the figures therefore share one color axis.

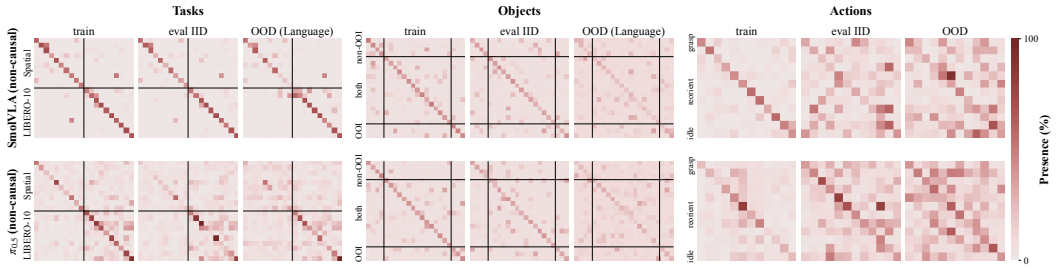


Figure 19: **Non-causal robot code signatures on held-out and paraphrased episodes.** Same layout as Figure 5, for SmolVLA / $\pi_{0.5}$ + AFTERIMAGE trained without the causal mask on the gather refiner. Task and object signatures are selected on in-distribution Non-causal rollouts and measured unchanged on held-out IID and Language-shift episodes from the same Non-causal eval.

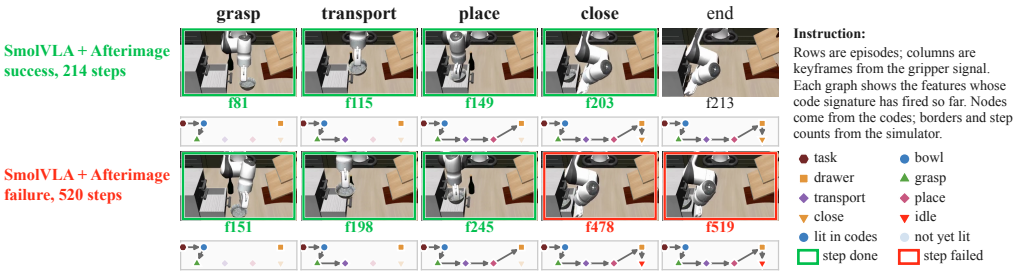


Figure 20: **Rollout progression under AFTERIMAGE (SmolVLA).** Same layout and reading key as Figure 6: filled = lit in codes; faded = not yet; on failure, the close node is replaced by a red idle node.

Qualitative rollout columns (Fig. 6). Columns are gripper-based keyframes (grasp / transport / place / close) for reading the episode layout; human phases follow Appendix I. Frame t shows the state before execution, and Figure 6 gives the filled / faded / success–failure reading key.

K.3 MORE ROLLOUTS, SUCCEEDED AND FAILED

Does the record predict the outcome? The code record is more useful for inspection when tied to the policy's behavior and not only to the instructed task. To test the relation, we score every Language-shift rollout of the evaluation runs of Table 2 by whether the instructed task's signature fires anywhere in its code record. We then cross-tabulate the score against the episode's outcome over the four suites and three runs, 6,000 episodes per backbone. On both backbones, episodes whose record carries the instructed signature succeed far more often than episodes without it. The record is therefore associated with the outcome and not with the instruction alone. On $\pi_{0.5}$, success is $96.7 \pm 0.6\%$ when the signature fires and $70.3 \pm 1.1\%$ when the signature is absent. On SmolVLA, success is $93.2 \pm 0.9\%$ against $52.5 \pm 1.3\%$. When the signature is absent, the odds of failure are 12.7 ± 2.0 times higher on $\pi_{0.5}$ and 12.6 ± 1.2 times higher on SmolVLA. The signature fires in $84.8 \pm 0.7\%$ of $\pi_{0.5}$ episodes and $74.8 \pm 0.8\%$ of SmolVLA episodes, and in $88.4 \pm 0.4\%$ and $84.1 \pm 0.4\%$ of the respective successful episodes. Pooled over runs, the cross-tabulation recovers the Language means of Table 2 for both backbones.

Figures 21–25 show further rollouts on the Spatial and LIBERO-10 suites reported in the main text, in the layout of Figure 6. Each figure takes one task with both successful and failed episodes and shows its episodes in order. **How to read.** Each row is one episode (green border = success, red = failure), and the five columns are keyframes along the episode. Under each AFTERIMAGE frame, a small graph marks the task's objects and the actions the episode has reached, and faded nodes have not been reached. A red $\times$ marks where a failed episode stopped (dropped object, still held, or never grasped). The graph is drawn from the rollout's phases and outcome and is not read from the codes. Object nodes are the instruction's goal targets, named in the legend. The only code-derived element is the task node in Figure 26, which names the task whose signature the codes fired, using the signatures of Figures 14a–17. Baselines have no code stream and show frames only (one success, one failure).

Table 12: **Swapping the task signatures of LIBERO-10 tasks 1 and 2, $\pi_{0.5}$ + AFTERIMAGE.** The two tasks share a scene and its objects, so each goal predicate is evaluable under the other's instruction and redirection is measurable rather than undefined. Percentage of episodes completing task 1, task 2, or neither, in distribution and under the Language shift. Feature-neutral replaces the signature with a motif tied to no task, and mean±std is over three runs. The pair is the hardest in the suite, so the unchanged row sits below the ten-task means of Table 11.

		In distribution			Language shift		
Instruction	Codes	Task 1	Task 2	Neither	Task 1	Task 2	Neither
Task 1	unchanged	**94.2**$\pm$3.8	0.0$\pm$0.0	5.8$\pm$3.8	**60.8**$\pm$7.2	0.0$\pm$0.0	39.2$\pm$7.2
Task 1	feature-neutral	12.0$\pm$2.0	0.0$\pm$0.0	88.0$\pm$2.0	8.3$\pm$3.8	0.0$\pm$0.0	91.7$\pm$3.8
Task 1	task-1 $\rightarrow$ task-2 signature	8.3$\pm$5.2	**25.8**$\pm$6.3	65.8$\pm$8.0	5.0$\pm$4.3	**21.7**$\pm$6.3	73.3$\pm$7.6
Task 2	unchanged	0.0$\pm$0.0	**90.1**$\pm$4.1	9.9$\pm$4.1	0.0$\pm$0.0	**78.9**$\pm$5.4	21.1$\pm$5.4
Task 2	feature-neutral	0.0$\pm$0.0	22.7$\pm$4.2	77.3$\pm$4.2	0.0$\pm$0.0	17.7$\pm$5.2	82.3$\pm$5.2
Task 2	task-2 $\rightarrow$ task-1 signature	**27.7**$\pm$6.4	14.2$\pm$4.9	58.2$\pm$7.1	**18.4**$\pm$5.6	11.6$\pm$4.5	70.1$\pm$6.8

Figure 26 examines a single episode whose codes name a task other than the instructed one, shown as a case rather than as a rate. The codes lie on the forward path, so the record names the task the fusion settled on whether or not the policy completes the episode. In the misrecognized episode, the policy acted on features selected for a different task. The wrong outcome therefore suggests a failure of fusion rather than of execution, although a single case does not establish the cause. Under exclusive routing, the next frozen layer computes its attention from the codes, so the record reads the contribution the policy acted on at the depth of the codes.

K.4 REDIRECTING THE POLICY BY SWAPPING TWO TASK SIGNATURES

Investigating whether a task signature steers the policy toward its task, we swap the signatures of LIBERO-10 tasks 1 and 2 on $\pi_{0.5}$ + AFTERIMAGE (Table 12). The two tasks share a scene and its objects, so each task's goal predicate can be satisfied under the other's instruction, and redirection is measurable rather than undefined. Writing the other task's signature in place redirects a minority of episodes to the target task in both directions and under paraphrase, whereas a feature-neutral replacement disrupts the policy without redirecting toward any task. The pair was chosen for the shared scene and is also the hardest in the suite, so the unchanged rows sit below the ten-task means of Table 11. The feature-neutral replacement lowers the instructed task to 12.0% and 22.7% in distribution and produces the other task in no episode. A motif tied to no task therefore disrupts without redirecting. The other task's signature instead produces the target task in 25.8% and 27.7% of episodes in distribution and 21.7% and 18.4% under the Language shift. In all four swap rows, the target task is completed more often than the instructed one. Measured against what the intact policy achieves on the target instruction, the redirected fraction averages 0.29. The four cells are indistinguishable at the present sample size, so the effect appears in both directions and persists under paraphrase. The two tasks share their objects, so the installed signature is unlikely to supply object identity. The result suggests the redirection is carried by the goal the installed signature names. Most remaining episodes complete neither task. The partial effect is consistent with an edit at one depth, since the layers below the edited depth are unmediated and the action expert attends to their keys and values. The other-task overwrite of Table 11, panel c, draws its replacement from a different suite whose objects are absent from the scene, so the overwrite can only degrade success. In the swap, the target task is achievable in the scene of measurement.

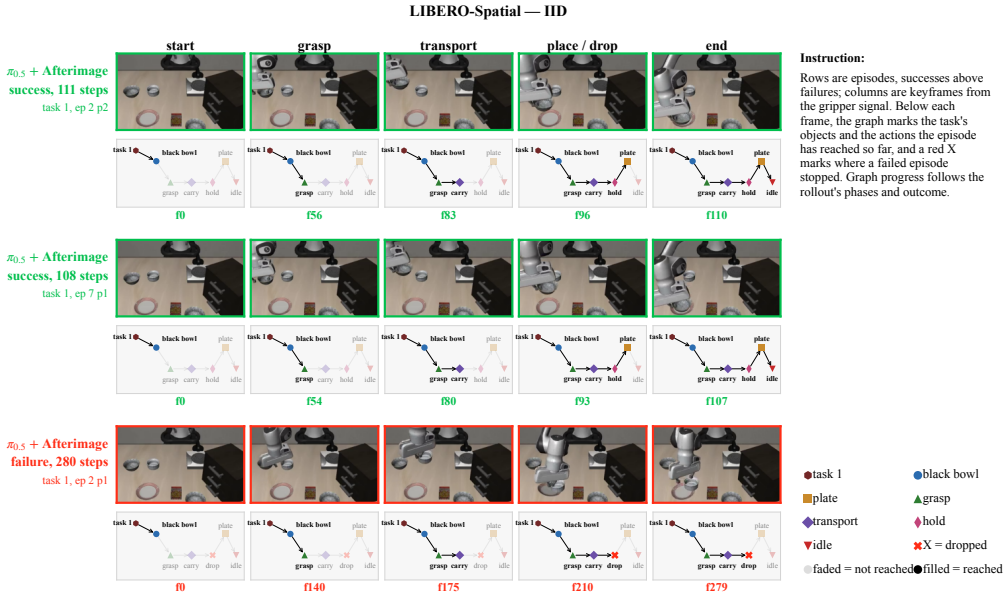


Figure 21: $\pi_{0.5}$ + **AFTERIMAGE on LIBERO-Spatial, held-out IID.** Rows are episodes of one task (green = success, red = failure), and columns are keyframes with step index f. The graph under each frame tracks the episode's phases and outcome, not the codes, and a red $\times$ marks where a failed episode stopped.

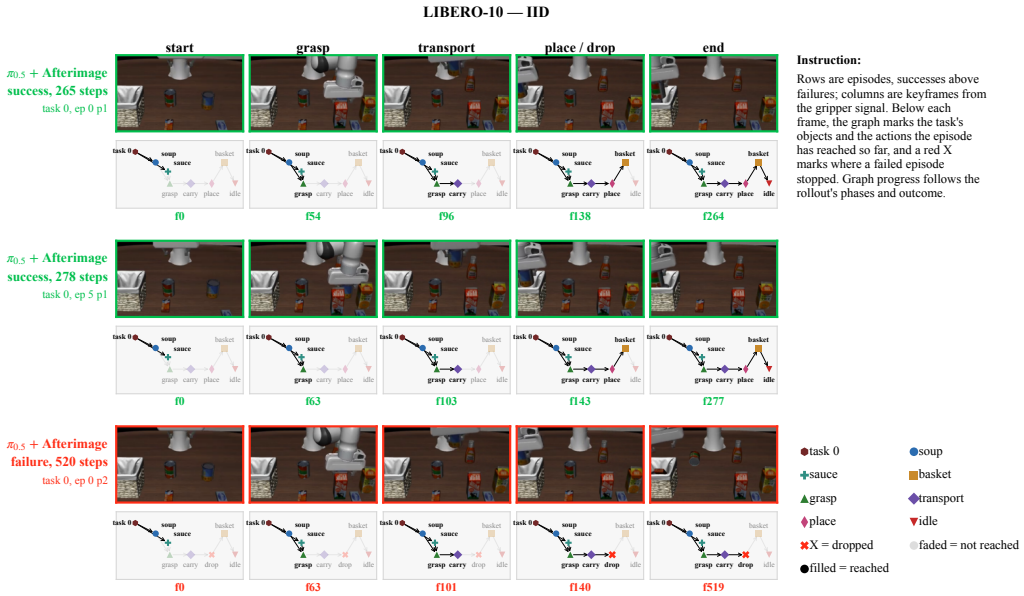


Figure 22: $\pi_{0.5}$ + **AFTERIMAGE on LIBERO-10, held-out IID.** Rows are episodes of one task (green = success, red = failure), and columns are keyframes with step index f. The graph under each frame tracks the episode's phases and outcome, not the codes, and a red $\times$ marks where a failed episode stopped.

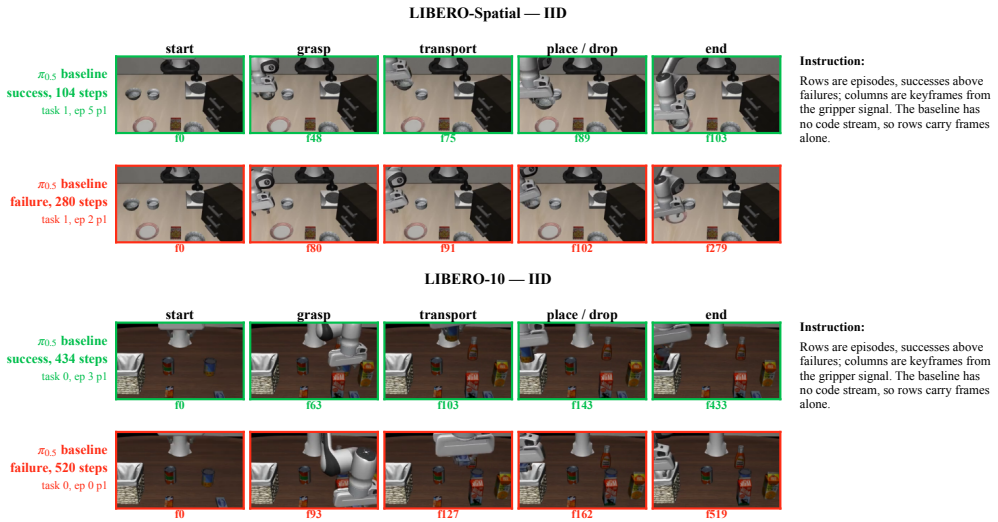


Figure 23: $\pi_{0.5}$ **baseline on LIBERO-Spatial (top) and LIBERO-10 (bottom), held-out IID.** Each panel shows one successful episode above one failed episode, and columns are the five phases. The number under each frame gives the step at which the frame was taken. The baseline emits no codes, so the rows carry frames alone.

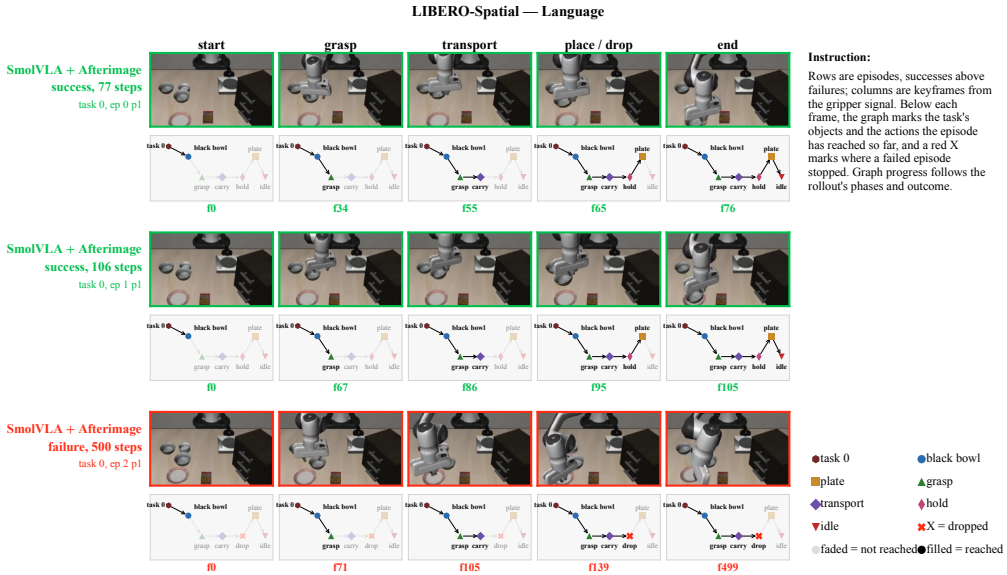


Figure 24: **SmolVLA + AFTERIMAGE on LIBERO-Spatial, Language shift.** Rows are episodes of one task (green = success, red = failure), and columns are keyframes with step index f. The graph under each frame tracks the episode's phases and outcome, not the codes, and a red $\times$ marks where a failed episode stopped.



Figure 25: **SmolVLA + AFTERIMAGE on LIBERO-10, Language shift.** Rows are episodes of one task (green = success, red = failure), and columns are keyframes with step index f. The graph under each frame tracks the episode's phases and outcome, not the codes, and a red $\times$ marks where a failed episode stopped.

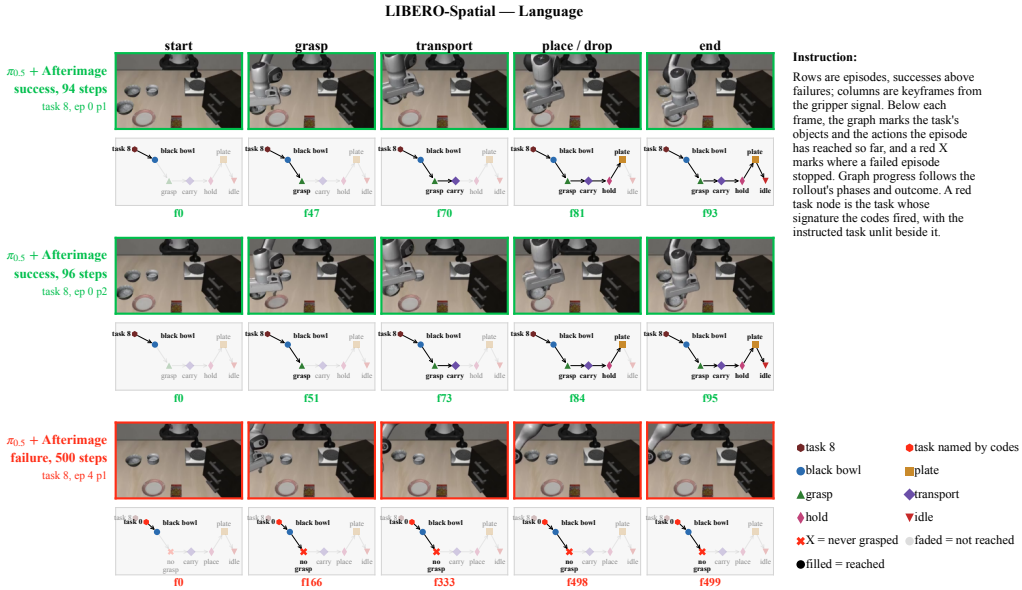


Figure 26: **Reading the record across repeats of one instruction.** $\pi_{0.5}$ + AFTERIMAGE on LIBERO-Spatial task 8 under the Language shift, given the same instruction three times. Two of the three rollouts carry the signature of task 8, while the third carries the signature of task 0 and no other. The record thus distinguishes rollouts sharing one instruction. The red task node is the task whose signature the codes fired, and the instructed task stays unlit beside it. The remaining graph nodes follow the rollout's phases and outcome.

L ABLATION DETAILS

This appendix extends the main-text ablations to every metric and suite. Figure 7 in the main text reports one metric per domain. The robot panel additionally includes a single-depth arm inserting AFTERIMAGE only at layer 16, the deepest depth of $\mathcal{S}$ (Language mean 77.0 ± 6.3 over the four suites). This appendix gives all three brain MRI metrics and every available robot result, covering the four in-distribution suites and all four suites of each of the three LIBERO-Pro axes for both backbones. CLEVRER is reported in the main text only.

Qualitative MRI under code destruction. Replacing codes at random or permuting their order collapses the synthesized volume into unstructured intensity. Figure 27 shows synthesized axial mid-slices under the two inference-time destructions of Section 3.4 for three synthesis conditions.

Brain MRI ablations on every metric. Figure 28 reports fidelity, sex accuracy, and age error together for the intact code, the two destructions, and the continuous bottleneck.

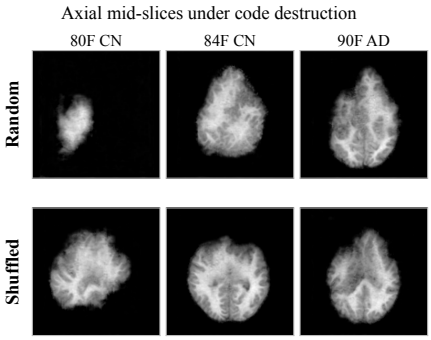


Figure 27: **Synthesized axial MRI under random and shuffled codes.** Columns are three synthesis conditions (80F CN, 84F CN, 90F AD); rows are the two inference-time destructions. Both destructions remove visible anatomical structure.

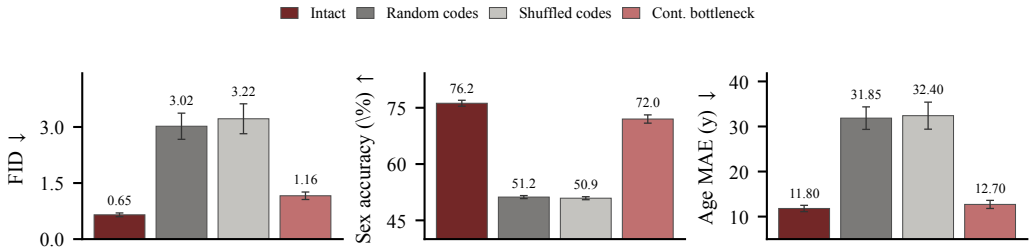


Figure 28: **Brain MRI ablations on all three metrics.** CN synthesis for the intact code, two inference-time destructions (average of three runs), and a continuous bottleneck in place of the quantizer. Lower FID and age MAE are better; higher sex accuracy is better. Both destructions return sex accuracy to chance and roughly triple age error. The continuous bottleneck degrades all three metrics far less than the destructions, although its scores still fall short of the intact run.

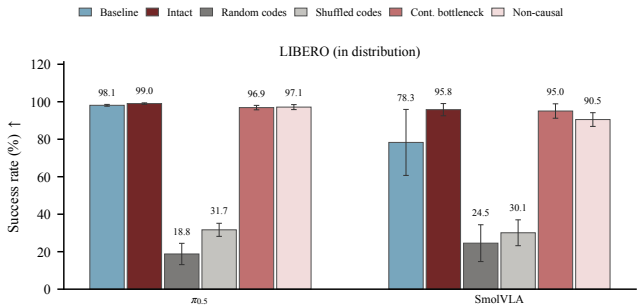


Figure 29: **Robot action planning ablations in distribution, both VLA backbones.** Each bar is the mean over the four LIBERO suites, and error bars give the standard deviation across suites; per-suite values are in the full robotics table. Replacing codes at random or permuting their order collapses success for both policies. In distribution, the continuous bottleneck stays within a few points of the intact code (and slightly below the $\pi_{0.5}$ backbone mean), whereas under distribution shift the intact code pulls ahead (Figure 30). Non-causal removes the causal gather mask.

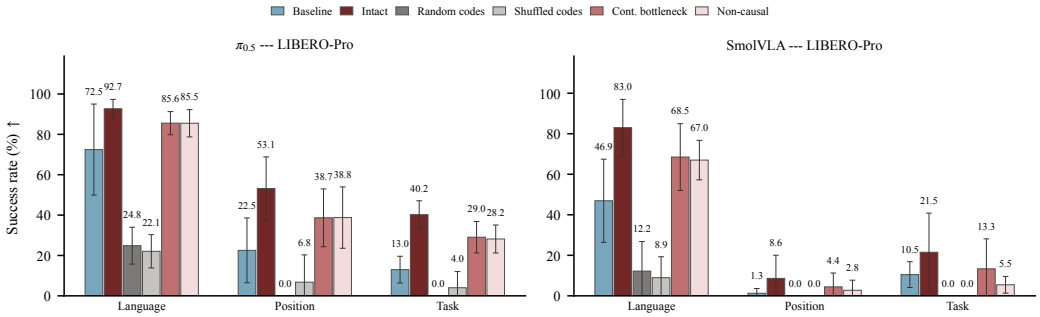


Figure 30: **Robot action planning ablations under distribution shift.** Each bar is the mean over the four suites of the shift axis, and error bars give the standard deviation across suites. Averaged over suites, AFTERIMAGE leads the backbone and the continuous bottleneck on every axis for both policies. Both destructions fall below the backbone on every axis and reach zero on most suites once objects move or the logic changes. Non-causal removes the causal gather mask.