DSCA: Dynamic Subspace Concept Alignment for Lifelong VLM Editing

Supplementary Material

7. Contents

- Theoretical Analysis of Non-Interference in DSCA
- Additional Methodology Details
- Evaluation Metrics
- Implementation Details and Hyperparameters
- Extended Experimental Results

8. Theoretical Analysis of Non-Interference in DSCA

8.1. Preliminaries

Let the frozen VLM encoder produce fused representations $\mathbf{h}_f \in \mathbb{R}^{d_f}$(as defined in Sec. 3.1). For each discovered concept C_k, DSCA maintains a low-dimensional semantic subspace with basis matrix $\mathbf{R}_k \in \mathbb{R}^{r_k \times d_f}$, where $r_k \ll d_f$.(Sec. 3.3) We view the rows of $\mathbf{R}_k$ as an orthonormal basis for the concept subspace

$$\mathcal{R}_k = \left\{ \mathbf{R}_k^\top \mathbf{u} \;\middle|\; \mathbf{u} \in \mathbb{R}^{r_k} \right\} \subset \mathbb{R}^{d_f}. \tag{11}$$

The corresponding orthogonal projector onto $\mathcal{R}_k$ is

$$\mathbf{P}_k = \mathbf{R}_k^\top \mathbf{R}_k \in \mathbb{R}^{d_f \times d_f}. \tag{12}$$

In Sec. 3.3, the intervention proposed by DSAM k is

$$\Psi_k(\mathbf{h}_f) = \mathbf{\Gamma}_k(\mathbf{h}_f)\, \mathbf{R}_k^\top \left[(\mathbf{W}_k \mathbf{h}_f + \mathbf{b}_k) - \mathbf{R}_k \mathbf{h}_f \right], \tag{13}$$

and the full edited representation (Sec. 3.5) is

$$\mathbf{h}'_f = \mathbf{h}_f + \sum_{k=1}^{K} w_k\, \Psi_k(\mathbf{h}_f), \tag{14}$$

where $\{w_k\}$ are routing weights (Sec. 3.4) and $\mathbf{\Gamma}_k(\mathbf{h}_f)$ is the diagonal gating matrix defined in Eq. 5.

For the theoretical analysis, it is convenient to absorb the component-wise gate into an effective subspace update. We therefore rewrite

$$\Psi_k(\mathbf{h}_f) = \mathbf{R}_k^\top\, \Delta_k(\mathbf{h}_f), \tag{15}$$

where $\Delta_k(\mathbf{h}_f) \in \mathbb{R}^{r_k}$ is an *effective* low-dimensional update that depends on $\mathbf{W}_k, \mathbf{b}_k$ and $\mathbf{\Gamma}_k(\mathbf{h}_f)$. This does not change the functional form of DSAM in practice, but lets us express the update as a sum of subspace-aligned residuals.

8.2. Assumptions

We now make explicit the structural assumptions under which non-interference guarantees hold.

A1. Orthogonal subspaces. The semantic subspaces are mutually orthogonal:

$$\mathbf{R}_i \mathbf{R}_j^\top = \mathbf{0} \quad \text{for all } i \neq j, \tag{16}$$

and each basis is row-orthonormal, $\mathbf{R}_k \mathbf{R}_k^\top = \mathbf{I}_{r_k}$. This implies that the projectors satisfy $\mathbf{P}_i \mathbf{P}_j = \mathbf{0}$ for $i \neq j$.

A2. Bounded routing and gating. For any $\mathbf{h}_f$, the routed update has bounded energy:

$$\sum_{k=1}^{K} |w_k|\, \|\Delta_k(\mathbf{h}_f)\|_2 \;\leq\; \Gamma_{\max}, \tag{17}$$

for some constant $\Gamma_{\max} > 0$.

A3. Concept-aligned training. Each DSAM k is trained only on samples assigned to concept C_k by the online clustering as in Sec. 3.2 . DSAM k learns to operate only within its own semantic region.

In practice, Assumption A1 is only approximately satisfied. The residualized Incremental PCA procedure described in Sec. 3.3 and detailed further in and detailed further in Sec.9.3, maintains small overlap between subspaces, which we quantify in Sec. 8.5.

8.3. Non-Interference Under Orthogonal Subspaces

We first state a clean result under exact orthogonality, then extend to the approximate case.

Lemma 1 (Non-interference for orthogonal subspaces)**.** *Let $\mathbf{h}_f \in \mathbb{R}^{d_f}$ and $\mathbf{h}'_f$ be related by Eq. 14. Suppose Assumption A1(in Sec.8.2) holds. Then, for any concept index i,*

$$\mathbf{P}_i \mathbf{h}'_f = \mathbf{P}_i \mathbf{h}_f + w_i\, \mathbf{P}_i \Psi_i(\mathbf{h}_f). \tag{18}$$

In particular, for all $j \neq i$, the contribution of Ψ_j vanishes in the projection onto $\mathcal{R}_i$:

$$\mathbf{P}_i \Psi_j(\mathbf{h}_f) = \mathbf{0} \quad \textit{for } j \neq i. \tag{19}$$

Proof. Using the effective subspace form $\Psi_k(\mathbf{h}_f) = \mathbf{R}_k^\top \Delta_k(\mathbf{h}_f)$ and Eq. 14,

$$\mathbf{h}'_f = \mathbf{h}_f + \sum_{k=1}^{K} w_k\, \mathbf{R}_k^\top \Delta_k(\mathbf{h}_f). \tag{20}$$

Projecting onto $\mathcal{R}_i$ gives

$$\mathbf{P}_i \mathbf{h}'_f = \mathbf{P}_i \mathbf{h}_f + \sum_{k=1}^{K} w_k\, \mathbf{P}_i \mathbf{R}_k^\top \Delta_k(\mathbf{h}_f) \tag{21}$$

$$= \mathbf{P}_i \mathbf{h}_f + \sum_{k=1}^{K} w_k\, \mathbf{R}_i^\top \left(\mathbf{R}_i \mathbf{R}_k^\top \right) \Delta_k(\mathbf{h}_f). \tag{22}$$

By Assumption A1(in Sec.8.2), $\mathbf{R}_i\mathbf{R}_k^\top = \mathbf{0}$ for all $k \neq i$, and $\mathbf{R}_i\mathbf{R}_i^\top = \mathbf{I}$. Therefore all cross-concept terms vanish and only the $k = i$ term remains:

$$\mathbf{P}_i\mathbf{h}'_f = \mathbf{P}_i\mathbf{h}_f + w_i\,\mathbf{R}_i^\top\Delta_i(\mathbf{h}_f) = \mathbf{P}_i\mathbf{h}_f + w_i\,\mathbf{P}_i\Psi_i(\mathbf{h}_f), \tag{23}$$

which proves both claims. □

Lemma 1 shows that, under ideal orthogonality, *edits in one semantic subspace cannot alter projections onto any other subspace*. In this sense, catastrophic interference is ruled out at the level of the subspace decomposition.

8.4. Approximate Orthogonality and Bounded Interference

In practice, residualized Incremental PCA produces subspaces that are only approximately orthogonal. We therefore consider the more realistic setting where inter-subspace overlap is small but non-zero.

Define the pairwise subspace overlap as

$$\mathrm{Overlap}(i,j) = \left\|\mathbf{R}_i\mathbf{R}_j^\top\right\|_F^2, \quad i \neq j, \tag{24}$$

and assume a uniform bound

$$\left\|\mathbf{R}_i\mathbf{R}_j^\top\right\|_F^2 \;\leq\; \varepsilon \quad \text{for all } i \neq j. \tag{25}$$

Corollary 1 (Bounded interference under ε-overlap)**.** *Suppose Assumptions A2(in Sec.8.2) and (25) hold. Let $\mathbf{h}'_f$ be given by Eq. 14. Then, for any concept index i, the interference from all other concepts in the projection onto $\mathcal{R}_i$ is bounded by*

$$\left\|\mathbf{P}_i(\mathbf{h}'_f - \mathbf{h}_f) - w_i\mathbf{P}_i\Psi_i(\mathbf{h}_f)\right\|_2 \;\leq\; \Gamma_{\max}\sqrt{\varepsilon}. \tag{26}$$

Proof. From the same expansion as in Lemma 1, the net contribution of all *other* concepts to the projection onto $\mathcal{R}_i$ is

$$\mathbf{r}_i = \sum_{j\neq i} w_j\,\mathbf{P}_i\Psi_j(\mathbf{h}_f) = \sum_{j\neq i} w_j\,\mathbf{R}_i^\top\left(\mathbf{R}_i\mathbf{R}_j^\top\right)\Delta_j(\mathbf{h}_f). \tag{27}$$

Using the triangle inequality and Cauchy–Schwarz,

$$\|\mathbf{r}_i\|_2 \leq \sum_{j\neq i} |w_j|\,\left\|\mathbf{R}_i^\top\left(\mathbf{R}_i\mathbf{R}_j^\top\right)\Delta_j(\mathbf{h}_f)\right\|_2 \tag{28}$$

$$\leq \sum_{j\neq i} |w_j|\,\left\|\mathbf{R}_i\mathbf{R}_j^\top\right\|_F\,\left\|\Delta_j(\mathbf{h}_f)\right\|_2 \tag{29}$$

$$\leq \sqrt{\varepsilon}\sum_{j\neq i} |w_j|\,\left\|\Delta_j(\mathbf{h}_f)\right\|_2. \tag{30}$$

By Assumption A2(in Sec.8.2), $\sum_j |w_j|\,\|\Delta_j(\mathbf{h}_f)\|_2 \leq \Gamma_{\max}$, which yields the desired bound. □

Corollary 1 implies that interference between concepts scales only with $\sqrt{\varepsilon}$, the square root of the maximum inter-subspace overlap. When subspaces are nearly orthogonal (small ε), the effect of other concepts on the projection of $\mathbf{h}_f$ onto $\mathcal{R}_i$ is provably small. This confirms the findings visualized in Fig.2(a)

8.5. Empirical Orthogonality Diagnostics

To verify that the assumptions above hold approximately in practice, we monitor the empirical subspace overlap throughout training. For a given checkpoint, we compute

$$\mathrm{MeanOverlap} = \frac{1}{K(K-1)}\sum_{i\neq j}\left\|\mathbf{R}_i\mathbf{R}_j^\top\right\|_F^2. \tag{31}$$

Here, K represents the total count of concept subspaces currently instantiated by the model, corresponding to the dynamic cluster set size defined in Sec.3.2. In our experiments, the residualized Incremental PCA procedure keeps $\mathrm{MeanOverlap}$ in the range of 10^{-3}, consistent with a small ε regime. Combined with the bounded-update assumption (A2,in Sec.8.2), this empirically supports the claim that DSCA edits are structurally localized and exhibit minimal cross-concept interference.

Discussion. The results above formalize the central intuition behind DSCA; by decomposing the fused representation space into (approximately) orthogonal concept-wise subspaces and restricting edits to operate within these subspaces, the effect of an edit becomes spatially localized in representation space. As a result, preserving previously learned concepts is no longer purely an algorithmic property (e.g., via replay or regularization), but also an *architectural* consequence of the subspace design.

9. Additional Methodology Details

9.1. Gating Implementation Details

As discussed in Sec. 3.3, the component-wise gating vector $\gamma_k(\mathbf{h}_f) \in [0,1]^{d_f}$ is implemented via a lightweight neural layer

$$\gamma_k(\mathbf{h}_f) = \sigma(W_{g,k}\mathbf{h}_f + b_{g,k}),$$

where σ is the element-wise sigmoid. To avoid quadratic parameter growth in d_f, we factorize $W_{g,k}$ as a low-rank bottleneck:

$$W_{g,k} = U_k V_k,$$

with $U_k \in \mathbb{R}^{d_f\times b}$, $V_k \in \mathbb{R}^{b\times d_f}$, and $b \ll d_f$. This ensures that each DSAM adds only $O(d_f b)$ parameters, allowing DSCA to scale to many concepts without a prohibitive memory footprint.

9.2. Exact Loss Definitions

For completeness, we provide the full mathematical form of the loss components introduced in Sec. 3.6. To navigate the trilemma of *task fidelity*, *locality*, and *cross-modal alignment*,(Sec.1, 3.1) we design a composite loss function. Our training strategy operates on batches containing both new "edit" samples ($\mathcal{D}_e$) and "out-of-scope" replay samples ($\mathcal{D}_o$). Each edit sample is denoted as (I_e, T_e), representing the input image and text for which the model's behavior is being updated. This dual-batch approach allows us to simultaneously learn the new task while preserving existing knowledge. The total loss is a weighted sum of four distinct objective terms.

9.2.1. Task Fidelity Loss ($\mathcal{L}_{\text{task}}$)

To ensure the edit is successful, we apply a standard causal language modeling loss to the edit batch $\mathcal{D}_e$. This loss minimizes the negative log-likelihood of the target token sequence $Y_{\text{target}} = (y_1, y_2, \ldots, y_L)$, conditioned on the input and our model's intervention:

$$\mathcal{L}_{\text{task}} = \mathbb{E}_{(\boldsymbol{I}_e, \boldsymbol{T}_e)\in\mathcal{D}_e}\left[\sum_{i=1}^{L} \text{CrossEntropy}(\text{Logits}_i, y_i)\right], \tag{32}$$

where Logits_i is the model's predicted logit distribution for the i-th token after the update h'_f has been applied. This directly optimizes the DSAM parameters to produce the correct textual output.

9.2.2. Cross-Modal Alignment Loss ($\mathcal{L}_{\text{align}}$)

An edit must not only produce the correct text but also maintain the VLM's fundamental cross-modal consistency. The update $\Delta\boldsymbol{h}_f$ is computed from the fused representation. To prevent this update from causing a modality drift, we enforce that the modified visual semantics remain coherent with the original textual semantics. We achieve this by regularizing the post-edit visual vector $\boldsymbol{h}'_{v,e} = \boldsymbol{h}_{v,e} + \Delta\boldsymbol{h}_{f,e}$ to remain aligned with the unmodified text vector $h_{t,e}$. This is formulated as a cosine similarity maximization, applied only to the edit batch $\mathcal{D}_e$:

$$\mathcal{L}_{\text{align}} = \mathbb{E}_{(\boldsymbol{I}_e, \boldsymbol{T}_e)\in\mathcal{D}_e}\left[1 - \frac{(\boldsymbol{h}'_{v,e})^\top \boldsymbol{h}_{t,e}}{\|\boldsymbol{h}'_{v,e}\|_2 \|\boldsymbol{h}_{t,e}\|_2}\right]. \tag{33}$$

This asymmetric application i.e. updating the visual modality to match the static text—acts as a stable anchor, preventing the model from altering the textual concept space during a visually-driven edit.

9.2.3. Contrastive Representation Distillation Loss ($\mathcal{L}_{\text{cdistill}}$)

To ensure locality, edits on one concept must not corrupt unrelated knowledge. A simple L2 distance loss on replay samples is insufficient as it penalizes all deviations equally and fails to preserve the *relational geometry* of the embedding space. We therefore employ a more powerful contrastive distillation loss on the replay batch $\mathcal{D}_o$. For each sample in $\mathcal{D}_o$, the updated representation $\boldsymbol{h}'_{f,o}$ should be far more similar to its own original version (computed by a frozen teacher model, $\boldsymbol{h}^{\text{frozen}}_{f,o}$) than to any other sample in the batch. This is formulated using an InfoNCE objective [33]:

$$\mathcal{L}_{\text{cdistill}} = -\mathbb{E}_{\boldsymbol{h}'_{f,o}\in\mathcal{D}_o}\left[\log \frac{\exp(\text{sim}(\boldsymbol{h}'_{f,o}, \boldsymbol{h}^{\text{frozen}}_{f,o})/\tau)}{\sum_{j\in\mathcal{D}_o}\exp(\text{sim}(\boldsymbol{h}'_{f,o}, \boldsymbol{h}^{\text{frozen}}_{f,o,j})/\tau)}\right], \tag{34}$$

where $\text{sim}(\cdot,\cdot)$ is the cosine similarity and τ is a temperature hyperparameter. This loss provides a rich training signal that explicitly safeguards the model's relational knowledge structure against catastrophic forgetting.

9.2.4. Gate Sparsity Loss ($\mathcal{L}_{\text{sparse}}$)

To enforce a strict locality principle, interventions must only occur when an input is confidently matched to a concept. For out-of-scope samples, the routing weights $\{w_k\}$ should ideally all be zero. We encourage this behavior by applying an L1 penalty to the weights for samples in the replay batch $\mathcal{D}_o$. The L1 norm is well-suited for this task as it promotes solutions where most weights are exactly zero, preventing spurious DSAM activations.

$$\mathcal{L}_{\text{sparse}} = \mathbb{E}_{\mathcal{D}_o}\left[\sum_{k=1}^{K} |w_k|\right]. \tag{35}$$

This regularizer ensures that interventions are triggered selectively, preserving the integrity of unrelated representations.

9.2.5. The Complete Objective Function

The final loss is a weighted sum of these four components:

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{align}}\mathcal{L}_{\text{align}} + \lambda_{\text{distill}}\mathcal{L}_{\text{cdistill}} + \lambda_{\text{sparse}}\mathcal{L}_{\text{sparse}}, \tag{36}$$

where the λ coefficients are hyperparameters that control the trade-off between plasticity and stability.

9.3. Dual-Mode Parameter Update Framework

A key aspect of DSCA's stability is the separation of how different components are updated. Model parameters are partitioned into two categories governed by different update mechanisms and schedules. This framework corresponds to the DSCA Continual Training Loop and procedures outlined in Algorithm 1 and 2.

Gradient-driven parameters (continuous update). This group contains all parameters directly responsible for executing the intervention. For each concept k (Sec. 3.2), this set includes the subspace transformation parameters

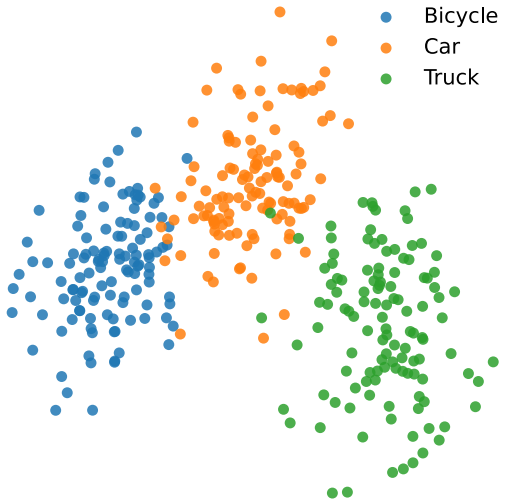

Figure 4. **Concept-wise subspace visualization.** t-SNE of fused representations projected through their assigned semantic subspaces $\mathbf{R}_k^\top \mathbf{h}_f$ for a subset of concepts (e.g., "car", "truck", "bicycle"). DSCA yields compact, well-separated clusters, indicating that edits remain confined to localized regions of the representation space.This empirically validates the conceptual illustration provided in Figure 1(d) of the main paper.

$\{\mathbf{W}_k, \mathbf{b}_k\}$ and gating parameters $\{\mathbf{W}_{g,k}, \mathbf{b}_{g,k}\}$ (Sec. 3.3). These parameters are learnable via backpropagation and are updated at every training step using gradients of the total loss $\mathcal{L}$ (Eq. 10).

Data-driven structural components (periodic update). This group contains the architectural backbone of the knowledge base; concept prototypes $\{\mathbf{p}_{k,v}, \mathbf{p}_{k,f}\}$ (Sec. 3.4, 3.2) and semantic subspace bases $\{\mathbf{R}_k\}$ (Sec. 3.3). These components are not updated via backpropagation but follow a data-dependent schedule:

- **Prototypes** ($\mathbf{p}_k$) are updated via Exponential Moving Average (EMA) whenever a new sample is assigned to their cluster.
- **Subspaces** ($\mathbf{R}_k$) follow a lifecycle based on data accumulation. A subspace is initialized via PCA only after its corresponding cluster buffer accumulates at least N_{min} samples, ensuring sufficient statistics to form valid principal components. Once active, the basis is refined periodically (every N_{refine} steps) using Incremental PCA on buffered features, with residualization across subspaces to maintain approximate orthogonality.

This dual-mode update separates rapid, fine-grained learning of "how to act" (DSAM parameters) from slower, structural re-organization of "how to represent concepts" (subspaces and prototypes), which is crucial for long-term stability and adaptability.

10. Evaluation Metrics

In this section, we provide the formal definitions of all evaluation metrics referenced in Sec. 4.2 of the main paper. Let f_{θ_0} denote the original (unedited) model and f_{θ_t} the model after t sequential edits. Each edit request is represented as a tuple $e = (v, p, o)$, consisting of a visual input v, textual prompt p, and desired target output o. The set of edits up to step t is denoted by $\mathcal{D}_{e,t}$. We use $\mathbb{I}\{\cdot\}$ for the indicator function.

10.1. Editing Metrics

To provide a comprehensive assessment of model editing performance, we evaluate each method along three fundamental axes: Efficacy, Generalization, and Locality. We formalize these metrics below. Please Let f_{θ_0} be the original, unedited model and f_{θ_t} be the model after t sequential edits. An edit request is a tuple $e = (v, p, o)$, consisting of a visual input v, a textual prompt p, and the desired target output o. Let $\mathcal{D}_{e,t}$ be the set of t edits applied to the model. We use $\mathbb{I}\{\cdot\}$ to denote the indicator function, which equals 1 if its condition is true, and 0 otherwise.

Efficacy: This measures the direct success of the edits.

- **Reliability (Rel.)** quantifies whether the model f_{θ_t} correctly produces the target output for all edit examples it has been trained on up to step t.

$$\text{Rel.} = \mathbb{E}_{(v,p,o)\sim\mathcal{D}_{e,t}} \left[\mathbb{I}\{f_{\theta_t}(v, p) = o\}\right] \tag{37}$$

Generalization: This assesses whether the edit was learned as a robust concept, rather than being merely memorized for one specific input.To test this, we evaluate the model on a set of inputs that are semantically equivalent but vary in their presentation. For a given input, we denote this set of variations as $\mathcal{N}(\cdot)$ and samples from it with a subscript g.

- **Textual Generalization (T-Gen.)** measures robustness to linguistic variation by evaluating paraphrases ($p_g \in \mathcal{N}(p)$) of the original prompt.

$$\text{T-Gen.} = \mathbb{E}_{(v,p,o)\sim\mathcal{D}_{e,t}} \mathbb{E}_{p_g\sim\mathcal{N}(p)} \left[\mathbb{I}\{f_{\theta_t}(v, p_g) = o\}\right] \tag{38}$$

- **Visual Generalization (V-Gen.)** measures robustness to visual variation by evaluating on images ($v_g \in \mathcal{N}(v)$) that depict the same subject but from different viewpoints or lighting conditions. This metric is also referred to as Modal Generality (M-Gen) in Table 2.

$$\text{V-Gen.} = \mathbb{E}_{(v,p,o)\sim\mathcal{D}_{e,t}} \mathbb{E}_{v_g\sim\mathcal{N}(v)} \left[\mathbb{I}\{f_{\theta_t}(v_g, p) = o\}\right] \tag{39}$$

Table 7. DSCA-specific hyperparameters used in all experiments. Values are kept fixed across editing and continual learning unless otherwise noted.

Setting	r	N_{min}	N_{refine}	τ	λ_{align}	λ_{distill}	λ_{sparse}
LLaVA-1.5-7B[15] (editing: E-VQA / E-IC / VLKEB)	128	32	500	0.07	0.5	1.0	1×10^{-2}
PaliGemma-3B[1] (CoIN continual learning[2])	128	32	500	0.07	0.5	1.0	1×10^{-2}

Locality: This ensures that edits do not degrade performance on unrelated inputs by comparing the behavior of f_{θ_t} to the original model f_{θ_0}. This is measured using a set of inputs that are unrelated to the edit context. We denote this set as $\mathcal{U}(\cdot)$ and samples from it with a subscript u.

- **Textual Locality (T-Loc.)** evaluates on unrelated, text-only inputs ($p_u \in \mathcal{U}(p)$).

$$\text{T-Loc.} = \mathbb{E}_{(v,p,o)\sim\mathcal{D}_{e,t}} \mathbb{E}_{p_u\sim\mathcal{U}(p)} [\mathbb{I}\{f_{\theta_t}(\emptyset, p_u) = f_{\theta_0}(\emptyset, p_u)\}] \tag{40}$$

- **Multimodal Locality (M-Loc.)** evaluates on unrelated multimodal inputs ($(v_u, p_u) \in \mathcal{U}(v, p)$).

$$\text{M-Loc.} = \mathbb{E}_{(v,p,o)\sim\mathcal{D}_{e,t}} \mathbb{E}_{(v_u,p_u)\sim\mathcal{U}(v,p)} [\mathbb{I}\{f_{\theta_t}(v_u, p_u) = f_{\theta_0}(v_u, p_u)\}] \tag{41}$$

Aggregate Score: Finally, to summarize overall performance, we compute the average score.

- **Average (Avg.)** provides a single, holistic measure by calculating the arithmetic mean of the five core metrics: Reliability, Textual Generalization, Visual Generalization, Textual Locality, and Multimodal Locality.

10.2. Continual Learning Metrics

To evaluate the performance of our method on the CoIN continual learning benchmark, we employ a set of standard metrics designed to measure catastrophic forgetting, knowledge transfer, and plasticity. Below, we provide the formal definitions for these metrics.

Let T be the total number of tasks in the continual learning sequence. We denote $A_{t,i}$ as the accuracy of the model on task i after it has been trained sequentially on tasks $1, 2, \ldots, t$. Consequently, $A_{i,i}$ is the accuracy on task i immediately after training on it, and $\bar{a}_i$ is the initial zero-shot accuracy on task i before any fine-tuning.

10.2.1. Average Accuracy (ACC)

Average Accuracy measures the final, overall performance of the model across all tasks after the entire sequence of T tasks has been learned. It provides a single score to summarize how well the model has retained knowledge and learned all tasks by the end.

ACC is calculated by averaging the final accuracies on each task i after the model has been trained on all T tasks.

$$\text{ACC} = \frac{1}{T}\sum_{i=1}^{T} A_{T,i} \tag{42}$$

10.2.2. Backward Transfer (BWT)

Backward Transfer measures the influence of learning a new task on the performance of previously learned tasks. It is the primary metric for quantifying catastrophic forgetting. A negative BWT value indicates forgetting, while a value close to zero implies knowledge retention.

BWT is calculated as the average difference between the final accuracy on a past task i ($A_{T,i}$) and the accuracy it had immediately after it was first learned ($A_{i,i}$).

$$\text{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1} (A_{T,i} - A_{i,i}) \tag{43}$$

10.2.3. Forward Transfer (FWT)

Forward Transfer measures the model's ability to generalize from past tasks to improve its learning on future, unseen tasks. It quantifies how much the knowledge gained from learning tasks 1 to $t-1$ helps the model's performance on the next task t before it is explicitly trained on it. A positive FWT is desirable.

FWT is calculated by averaging the difference between the model's accuracy on a new task i before training on it ($A_{i-1,i}$) and its initial zero-shot accuracy on that same task ($\bar{a}_i$).

$$\text{FWT} = \frac{1}{T-1}\sum_{i=2}^{T} (A_{i-1,i} - \bar{a}_i) \tag{44}$$

10.2.4. Average Task Accuracy (A_t)

Average Task Accuracy assesses the model's plasticity—its ability to effectively learn each new task. It measures the peak performance achieved on each task right after its respective training phase is complete, without considering whether that knowledge is later forgotten.

It is the average of the accuracies on each task i measured immediately after the model has finished training on that task.

$$A_t = \frac{1}{T}\sum_{i=1}^{T} A_{i,i} \tag{45}$$

11. Implementation Details and Hyperparameters

In this section, we detail the experimental setup and hyperparameter configurations used to train and evaluate DSCA. All experiments were conducted using the PyTorch framework on $8\times$ NVIDIA A100 (80GB) GPUs with mixed-precision training.

Backbone Models. We apply DSCA to two distinct vision-language architectures to demonstrate generality:

- **LLaVA-1.5-7B [15]:** Used for all model editing benchmarks (E-VQA[5], E-IC[5], VLKEB[11]).
- **PaliGemma-3B [1]:** Used for the CoIN continual learning benchmark[2].

Hyperparameter Configuration. To ensure reproducibility and fair comparison, we maintain a fixed set of hyperparameters across all benchmarking experiments. These specific values are listed in **Table 7**.

Key hyperparameters include:

- **Subspace Rank (r):** The dimensionality of the semantic subspaces ($r = 128$). (Sec.3.3)
- **Subspace Initialization Threshold (N_{min}):** The minimum number of samples a concept cluster must accumulate before its semantic subspace basis $\mathbf{R}_k$ is initialized via PCA and the corresponding DSAM becomes active ($N_{\text{min}} = 32$). (Algo.2, Sec.9.3)
- **Refinement Interval (N_{refine}):** The frequency of Incremental PCA updates for the semantic bases ($N_{\text{refine}} = 500$ steps).(Algo.1, Sec.9.3)
- **Routing Temperature (τ):** Controls the sharpness of the routing distribution ($\tau = 0.07$). (Sec.3.4)
- **Loss Coefficients (λ):** We balance the objective using $\lambda_{\text{align}} = 0.5$ (cross-modal alignment), $\lambda_{\text{distill}} = 1.0$ (contrastive distillation), and $\lambda_{\text{sparse}} = 1 \times 10^{-2}$ (gate sparsity). (Eq.10). This value of $\lambda_{\text{sparse}} = 1\times10^{-2}$ is reflected as the Operating Point in Graph 3(b)

Unless otherwise noted in the specific experimental subsection, these values remain constant throughout the lifelong editing and continual learning processes.

12. Extended Experimental Results

We provide expanded comparisons against a wider range of baselines in Tables 8, 9, and 10.

12.1. Expanded Single-Edit Performance

Table 8 provides a comprehensive single edit success comparison on the E-VQA and E-IC benchmarks.

All baseline numbers, including standard fine-tuning variants and retrieval-based methods, are sourced from **DualEdit [30]**. Results for **FT-V** (Cheng et al., 2023), **FT-L** (Cheng et al., 2023), **KE** (De Cao et al., 2021), **IKE** (Li et al., 2023), **TP** (Huang et al., 2023), and **VisEdit**[4] should be cross-referenced with the experiment tables provided in **DualEdit [30]**.

12.2. Expanded Lifelong Editing Performance

Table 9 details performance after $t = 1,000$ sequential edits on LLaVA-1.5-7B.

All baseline numbers are sourced from **LiveEdit [3]**. Results for **RECIPE**(Chen et al. 2024), **LEMoE**(Wang et al. 2024), **FT-M** (Cheng et al. 2023), and **FT-L** (Cheng et al. 2023) are taken directly from the Sequential Editing benchmarks reported in **LiveEdit [3]**.

12.3. Expanded Continual Learning on CoIN

Table 10 reports results on the CoIN benchmark using the PaliGemma-3B backbone[1].All baseline numbers are sourced from **PAM [32]**. To establish performance bounds, we include three foundational setups defined in their work:

- **Zero-shot:** Evaluates the pre-trained PaliGemma base model directly without further training. This assesses the model's inherent generalization capabilities prior to task-specific adaptation.
- **Independent:** Fine-tunes a separate LoRA module for each task individually. This serves as an upper bound for task-specific performance by completely isolating tasks to prevent interference, though it requires maintaining distinct parameters for every task.
- **Multitask:** Trains a single LoRA module on all tasks concurrently. This represents the theoretical upper bound for a single shared model by assuming simultaneous access to all datasets, though it violates the sequential data constraints of Continual Learning.

Results for **LwF** (Li and Hoiem 2017), **I-LoRA** (Li et al. 2025), **O-LoRA** (Wang et al. 2023), **MoELoRA**[2], and **MagMax [22]** are cited as reported in the Experiments section of **PAM [32]**. Please refer to **PAM [32]** for the specific implementation details of these continual learning baselines.

Table 8. Extended Single Edit success comparision. Baselines from DualEdit[30]

Methods	E-VQA[5]						E-IC[5]					
	Rel.	**T-Gen.**	**V-Gen.**	**T-Loc.**	**M-Loc.**	**Avg.**	**Rel.**	**T-Gen.**	**V-Gen.**	**T-Loc.**	**M-Loc.**	**Avg.**
LLaVA-V1.5 [15]												
FT-V	31.68	29.96	26.68	100.00	91.23	55.91	52.85	51.57	48.63	100.00	92.55	69.12
FT-L	31.78	30.02	26.91	99.94	92.03	56.14	53.00	51.02	49.29	98.91	94.89	69.42
KE	85.86	84.00	82.23	93.57	73.06	83.74	83.54	82.15	81.12	92.46	73.83	82.62
IKE	91.35	90.84	91.08	60.18	51.08	76.91	93.72	88.37	76.99	76.60	64.90	80.12
SERAC [26]	82.51	81.60	80.05	100.00	57.48	80.33	43.08	42.37	42.85	100.00	7.63	47.19
MEND [25]	92.30	92.16	92.10	90.30	81.13	89.60	93.76	93.46	92.14	91.60	87.59	91.71
TP	38.68	36.27	31.26	95.31	91.41	58.59	59.07	57.01	55.51	64.79	89.26	65.13
LTE [13]	94.16	93.54	93.06	83.76	81.65	89.23	93.60	92.38	91.18	85.54	88.49	90.24
VisEdit [4]	95.78	94.21	94.37	100.00	91.11	95.09	95.06	94.87	94.35	100.00	95.23	95.90
DualEdit [30]	96.94	96.43	96.20	100.00	99.61	97.84	96.76	96.52	96.24	100.00	99.74	97.85
DSCA (Ours)	**98.12**	**97.30**	**97.25**	**100.00**	**99.83**	**98.50**	**98.00**	**97.10**	**97.02**	**100.00**	**99.90**	**98.00**

Table 9. Expanded Lifelong editing results ($t = 1000$ edits) on LLaVA-1.5-7B [15]. Baselines from LiveEdit [3].

Methods	E-VQA[5]						VLKEB[11]					
	Rel.	**T-Gen.**	**V-Gen.**	**T-Loc.**	**M-Loc.**	**Avg.**	**Rel.**	**T-Gen.**	**V-Gen.**	**T-Loc.**	**M-Loc.**	**Avg.**
LLaVA-V1.5 [15]												
LiveEdit[3]	92.93	90.16	84.30	100.00	96.43	92.76	92.22	83.97	82.75	100.00	100.00	91.79
LTE [13]	83.93	82.55	81.34	83.97	73.09	80.98	64.51	56.26	64.80	80.85	76.52	68.59
MEND[25]	0.04	0.05	0.05	0.08	0.09	0.06	0.03	0.05	0.07	0.06	0.08	0.06
SERAC [26]	85.57	75.58	82.01	62.46	15.69	64.26	60.93	56.49	60.06	52.94	15.04	49.09
FT-L	71.39	59.83	57.41	55.55	48.99	58.63	68.14	66.38	66.98	65.61	75.35	68.49
FT-M	69.57	56.34	44.07	100.00	41.47	62.29	53.41	48.80	43.16	100.00	57.03	60.48
TP	16.56	16.80	15.65	7.28	15.60	14.38	5.46	4.81	5.51	2.77	7.19	5.15
RECIPE	87.00	76.81	83.09	86.95	87.03	84.18	62.00	56.84	61.50	85.37	82.07	69.56
LEMoE	30.80	25.75	24.32	71.45	46.23	39.71	67.97	61.07	58.16	48.48	44.06	55.95
DSCA (ours)	**96.85**	**93.10**	**88.00**	**100.00**	**98.20**	**95.23**	**98.10**	**93.80**	**89.70**	**100.00**	**100.00**	**96.72**

Table 10. Expanded table for perfomance on COIN benchmark[2]. Higher is better for all metrics; less negative BWT indicates less forgetting.Baselines are taken from PAM[32]

METHOD	**ACC**	**BWT**	**FWT**	A_t
ZERO-SHOT	24.74	-	-	-
INDEPENDENT	76.46	-	-	-
MULTITASK	73.93	-	-	-
FINE-TUNE	43.36±8.18	-39.51±9.87	7.71±2.51	76.29±0.18
LWF	47.15±3.52	-33.66±3.84	9.67±1.22	75.20±0.32
I-LoRA	42.11±6.55	-40.95±8.13	7.89±2.60	76.24±0.35
O-LoRA	46.53±6.88	-32.08±7.87	9.45±3.02	73.27±1.01
MoELoRA [2]	46.59±9.98	-36.40±11.97	7.79±2.24	**76.93±0.27**
MagMax[22]	45.74±0.88	-22.68±6.51	4.75±3.36	76.29±0.18
PAM[32]	49.89±1.66	-19.45±0.95	**11.11±0.09**	76.31±0.03
DSCA (ours,with PaliGemma-3B[1])	**49.96±0.72**	**-9.37±1.02**	11.04±0.13	76.48±0.07