BEYOND FIXED TARGETS: CONTEXT-ADAPTIVE SUBSPACE NAVIGATION FOR MITIGATING OBJECT HALLUCINATIONS IN LVLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) generate fluent responses but often hallucinate objects that are absent from the input image. Existing subspace interventions reduce hallucinations by moving subspace components toward a target that remains fixed across inputs, but this uniform correction can also suppress mentions of real objects, creating a trade-off between hallucination reduction and object recall. To address this challenge, we propose **S**ubspace-**C**onstrained **A**nchor **N**avigation, termed SCAN, which retains a shared subspace but replaces the fixed target with an anchor adapted to each context. Specifically, SCAN separates the hidden state into a component within the hallucination-associated subspace and a residual orthogonal to it. Using only this residual as context, a lightweight Navigator predicts an anchor and per-direction strengths that control how far the subspace component moves toward the anchor, leaving the residual unchanged. With the LVLM backbone and subspace basis frozen, the Navigator is trained on preference pairs for captioning and object existence judgments to discourage hallucinated mentions while preserving object recall. Extensive experiments across multiple tasks and LVLM backbones demonstrate that SCAN consistently reduces object hallucinations and achieves a favorable hallucination–recall trade-off while preserving decoding efficiency.

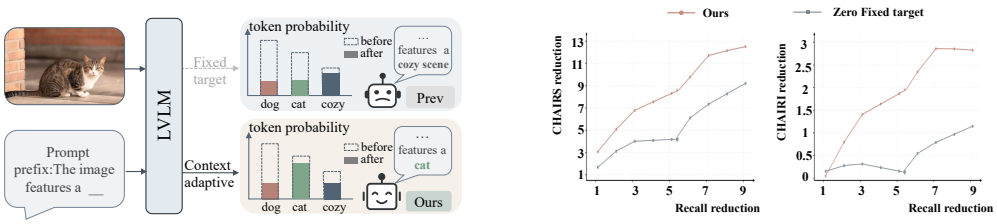
1 INTRODUCTION

Large vision-language models (LVLMs) (Liu et al., 2025b; Bai et al., 2025a) have demonstrated impressive performance across various tasks, including image captioning, visual question answering (Bai et al., 2025b), and multimodal reasoning (Huang et al., 2026). Despite these advances, LVLMs remain vulnerable to object hallucination (Rohrbach et al., 2018; Li et al., 2023b; Kaul et al., 2024), producing fluent and plausible responses that mention objects absent from the input image. Such unsupported object mentions undermine the reliability of generated descriptions (Chen et al., 2026), making their mitigation important for reliable use in practical applications.

Existing approaches to mitigating object hallucination include backbone fine-tuning and inference-time intervention. Backbone fine-tuning updates the LVLM through instruction tuning (Liu et al., 2024a), contrastive learning (Jiang et al., 2024), or preference optimization (Zhao et al., 2025). Although effective, these methods can require substantial computational resources to optimize large backbones. In contrast, inference-time interventions keep the backbone frozen and modify internal activations during generation to mitigate hallucinations (Liu et al., 2025a; Su et al., 2025). Among these approaches, subspace interventions constrain hidden state updates to a compact set of directions associated with hallucination, enabling efficient correction during generation (Lin et al., 2026; Dastmalchi et al., 2026).

Despite their efficiency, fixed-target subspace interventions move hidden state components toward the same target regardless of the image, prompt, or generated text. As illustrated in Figure 1(a), this uniform correction can suppress mentions of real objects along with hallucinated ones, creating a trade-off between hallucination reduction and object recall. Our approach improves this trade-off by achieving greater hallucination reduction than the zero-target baseline at matched recall loss, as shown in Figure 1(b). *Formally, under a squared error criterion for target approximation, even the*

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.



(a) Fixed versus adaptive targets.

(b) Hallucination–recall trade-off.

Figure 1: SCAN replaces a fixed target with an anchor adapted to each context, achieving greater hallucination reduction at matched recall loss. Recall loss denotes the absolute decrease from Vanilla.

best fixed target incurs an irreducible gap if the optimal target varies with context. This mismatch, which we term **fixed-target mismatch**, highlights that sharing a subspace does not justify sharing a single target across contexts. Together, these findings motivate retaining the shared subspace while replacing the fixed target with an anchor adapted to each context.

Motivated by these findings, we propose a **S**ubspace-**C**onstrained **A**nchor **N**avigation framework for mitigating object hallucination in LVLMs, termed SCAN, which retains a shared subspace while adapting the correction target to each context. Since corrections are confined to the hallucination-associated subspace, the residual orthogonal to this subspace remains unchanged by each intervention. We use this preserved residual as the context for target selection, excluding the component being corrected from the prediction input. Specifically, a lightweight Navigator maps the residual to an anchor and per-direction strengths, which determine where and how far to move the subspace component. To learn corrections that suppress hallucinated mentions while preserving object recall, we train the Navigator on preference pairs for captioning and object existence judgments, keeping the backbone and subspace basis frozen. At inference, the learned correction is applied within the existing backbone forward pass, retaining one pass per decoding step. Our main contributions are summarized as follows:

- We identify and analyze fixed-target mismatch in subspace interventions. Under a squared error criterion for target approximation, we show that even the best fixed target incurs an irreducible gap when the optimal target varies with context.
- We propose SCAN, which uses the orthogonal residual to predict an anchor and navigation strengths adapted to each context. By confining updates to the shared subspace, SCAN preserves the residual while learning how to correct the subspace component through a lightweight Navigator with the backbone frozen.
- Extensive experiments across models and tasks demonstrate that SCAN consistently reduces object hallucinations and achieves a favorable hallucination–recall trade-off. Additional efficiency measurements show that SCAN retains decoding throughput close to that of the original backbone.

2 Related Work

2.1 Hallucination Mitigation in LVLMs

To mitigate hallucinations in LVLMs, various approaches have been proposed. Among them, backbone fine-tuning approaches such as LRV-Instruction (Liu et al., 2024a) and HA-DPO (Zhao et al., 2025) improve visual grounding through instruction tuning and preference optimization, respectively. While effective, these methods require additional data construction and backbone updates. To avoid backbone updates, inference-time interventions modify internal activations, as exemplified by VTI (Liu et al., 2025a) and Activation Steering Decoding (Su et al., 2025). Related subspace methods suppress selected components through weight projection in Nullu (Yang et al., 2025) or hidden state projection in CIPHER (Dastmalchi et al., 2026). To adapt this suppression to the current context, HulluEdit (Lin et al., 2026) adjusts both the subspaces and suppression strengths while preserving visual evidence. However, these projection and shrinkage mechanisms still move selected

components toward zero, leaving open whether a different target would better suit each context. In contrast, SCAN uses the orthogonal residual to predict an anchor and per-direction strengths within a shared subspace, learning both where and how strongly to correct its components.

2.2 Representation-Space Interventions

Representation-space interventions control model behavior by modifying internal activations while keeping the backbone frozen. Early approaches such as ITI (Li et al., 2023a) and CAA (Rimsky et al., 2024) extract steering directions from data and apply them during inference. Moving beyond extracted directions, ReFT (Wu et al., 2024) learns task-specific interventions, with LoReFT jointly learning a low-rank subspace and a mapping from hidden states to target coordinates. Other approaches adapt interventions through their dynamics, direction, or strength. ODESteer (Zhao et al., 2026) uses barrier functions to guide multi-step, adaptive steering through ODE dynamics, while MARI (Yu et al., 2026) adapts intervention directions and strengths across samples and uses energy-based gating to determine when to intervene. At the token level, PSR (Heyman & Vandeputte, 2026) predicts steering coefficients from activations by learning to imitate interventions induced by prompting. In contrast, SCAN uses the preserved orthogonal residual as context to learn an adaptive anchor and per-direction navigation strengths within a fixed hallucination-associated subspace.

2.3 Summary and Motivation

Taken together, these studies highlight the importance of both constraining where an intervention acts and adapting how it changes the representation. Building on these insights, we focus on selecting a target suited to each context within a shared hallucination-associated subspace. To this end, we analyze fixed-target mismatch and develop SCAN, which uses the orthogonal residual to guide the selection of an appropriate target.

3 Methodology

This section presents the formulation and implementation of SCAN. We first analyze the approximation gap between fixed targets and the optimal target conditioned on context. We then introduce the subspace navigation mechanism, explaining how the orthogonal residual guides target prediction and how updates remain within the shared subspace. Finally, we describe the offline construction of the subspace and the training of the Navigator using preference pairs.

3.1 Fixed-Target Mismatch

To motivate adapting the correction target to context, we formalize fixed-target mismatch. Specifically, we analyze the approximation error incurred by using a single target across decoding contexts.

At decoding step t, let $h_t \in \mathbb{R}^d$ denote the current token's hidden state at the selected intervention layer. Let $B \in \mathbb{R}^{d\times k}$ be a fixed, backbone-specific basis of the hallucination-associated subspace, with $B^\top B = I_k$. Its construction is described in Section 3.3. We decompose the hidden state as

$$\begin{aligned} c_t &= B^\top h_t, & z_t &= h_t - Bc_t, \\ h_t &= Bc_t + z_t, & B^\top z_t &= 0. \end{aligned} \tag{1}$$

Here, $c_t \in \mathbb{R}^k$ contains the current subspace coordinates, and $z_t \in \mathbb{R}^d$ is the residual orthogonal to the subspace. We express correction targets in the same coordinate space as c_t and use z_t as the context for target selection.

To isolate target selection, we hold the intervention layer and basis B fixed and abstract away intervention strength and the update rule. Under a fixed distribution of decoding contexts, let Z denote the random counterpart of z_t, and let $C^\dagger \in \mathbb{R}^k$ represent the desired anchor for the sampled context. This anchor is an analysis variable used to define the target approximation problem. Assuming $\mathbb{E}[\|C^\dagger\|_2^2] < \infty$, for any measurable rule $a(Z)$ with finite second moment, define

$$\mathcal{R}(a) = \mathbb{E}\left[\|C^\dagger - a(Z)\|_2^2\right], \qquad \mu(Z) = \mathbb{E}[C^\dagger \mid Z]. \tag{2}$$

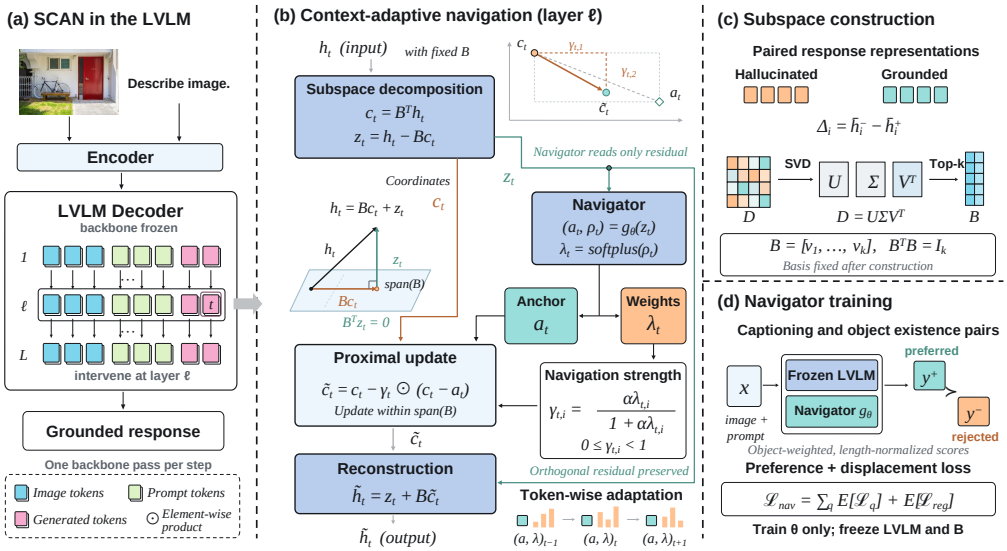


Figure 2: Overview of SCAN. (a) Frozen LVLM, single-pass intervention. (b) The Navigator uses only the residual z_t to predict an adaptive anchor and per-direction weights that steer c_t while preserving z_t. (c) SVD constructs frozen B from paired differences. (d) Only the Navigator is trained.

The conditional mean $\mu(Z)$ provides the optimal approximation based on the residual.

Lemma 1 (Conditional optimality)**.** *For every admissible target-selection rule a,*

$$\mathcal{R}(a) = \mathcal{R}(\mu) + \mathbb{E}\left[\|a(Z) - \mu(Z)\|_2^2\right]. \tag{3}$$

Thus, μ uniquely minimizes $\mathcal{R}$ among rules based on Z, up to almost-sure equality.

The decomposition follows by expanding the squared error and using $\mathbb{E}[C^\dagger - \mu(Z) \mid Z] = 0$ to cancel the cross term. For a constant target b shared across contexts, the excess risk is therefore its expected squared distance from $\mu(Z)$, motivating the following definition of fixed-target mismatch.

Definition 1 (Fixed-target mismatch)**.** For a fixed target $b \in \mathbb{R}^k$, define

$$\begin{aligned} \mathcal{M}(b) &= \mathbb{E}\left[\|\mu(Z) - b\|_2^2\right], \\ \mathcal{M}_{\text{fix}} &= \min_{b \in \mathbb{R}^k} \mathcal{M}(b). \end{aligned} \tag{4}$$

By Lemma 1, $\mathcal{M}(b) = \mathcal{R}(b) - \mathcal{R}(\mu)$. The following proposition separates the mismatch into variation across contexts and deviation from the best fixed target.

Proposition 1 (Fixed-target approximation gap)**.** *Let $\bar{\mu} = \mathbb{E}[\mu(Z)]$ and define the contextual target variation as $V_{\text{ctx}} = \mathbb{E}[\|\mu(Z) - \bar{\mu}\|_2^2]$. For any fixed target $b \in \mathbb{R}^k$,*

$$\mathcal{M}(b) = \mathcal{R}(b) - \mathcal{R}(\mu) = V_{\text{ctx}} + \|b - \bar{\mu}\|_2^2, \tag{5}$$

where $V_{\text{ctx}} = \text{Tr}(\text{Cov}[\mu(Z)])$.

The unique best fixed target is $b = \bar{\mu}$, and $\mathcal{M}_{\text{fix}} = V_{\text{ctx}}$. Hence, the irreducible mismatch is strictly positive whenever $\mu(Z)$ is not almost surely constant.

The two terms identify distinct limitations. Choosing a better fixed target can reduce the offset $\|b - \bar{\mu}\|_2^2$, but it cannot remove V_{ctx}, which captures variation in the conditional mean target across contexts. Thus, even an optimally chosen fixed target retains an approximation gap when this conditional mean varies.

Corollary 1 (Zero-target specialization)**.** *The zero target is optimal within the fixed-target class if and only if $\bar{\mu} = \mathbf{0}_k$. Its mismatch is*

$$\mathcal{M}(\mathbf{0}_k) = V_{\text{ctx}} + \|\bar{\mu}\|_2^2.$$

Relative to the best fixed target, using zero therefore incurs an additional offset of $\|\bar{\mu}\|_2^2$. Optimizing the fixed target can eliminate this offset but cannot reduce V_{ctx}.

A learned anchor can exploit this variation, provided that it approximates the conditional mean sufficiently well. The following criterion makes this requirement explicit.

Corollary 2 (Context-dependent approximation criterion)**.** *For any admissible rule $\widehat{a}(Z)$,*

$$\mathcal{R}(\widehat{a}) - \min_{b\in\mathbb{R}^k} \mathcal{R}(b) = \mathbb{E}\Big[\|\widehat{a}(Z) - \mu(Z)\|_2^2\Big] - V_{\mathrm{ctx}}. \tag{6}$$

Consequently, $\widehat{a}(Z)$ outperforms every fixed target under the squared error criterion if and only if its approximation error relative to $\mu(Z)$ is below V_{ctx}.

These results motivate retaining the shared subspace while learning an anchor conditioned on the orthogonal residual. In SCAN, this anchor is predicted by the Navigator introduced next and learned through preference pairs rather than direct supervision of $C^\dagger$. Appendix B provides full proofs and further analysis of conditioning information.

3.2 Context-Adaptive Subspace Navigation

Subspace Decomposition. SCAN uses subspace decomposition to confine corrections to hallucination-associated directions while preserving a context representation for target prediction. At each autoregressive step, Eq. (1) separates the current token's hidden state into the subspace component Bc_t and the orthogonal residual z_t. This orthogonality ensures that updates to the coordinates c_t affect only the subspace component, leaving z_t unchanged at the current token and layer. The preserved residual therefore serves as the Navigator's context input, allowing target prediction without directly using the coordinates being corrected.

Anchor Navigation. SCAN uses the residual to predict an anchor and direction-wise weights, specifying the correction target and how strongly each subspace coordinate is encouraged to approach it. These predictions come from a lightweight Navigator g_θ, shared across decoding steps:

$$\begin{aligned}(a_t, \rho_t) &= g_\theta(z_t),\\ \lambda_t &= \mathrm{softplus}(\rho_t).\end{aligned} \tag{7}$$

Here, $a_t \in \mathbb{R}^k$ specifies the target coordinates, while $\rho_t \in \mathbb{R}^k$ contains unconstrained outputs that softplus maps to positive weights $\lambda_t \in \mathbb{R}^k_{>0}$. Assigning a separate weight to each direction allows different coordinates to be encouraged toward their targets to different degrees. Although the Navigator parameters are shared across steps, changes in z_t allow the anchor and weights to adapt to the decoding context. These outputs guide the proximal update described next.

Proximal Update. SCAN uses a proximal update to move the subspace coordinates toward the predicted anchor while balancing the weighted attraction against deviation from their current values. With $\Lambda_t = \mathrm{diag}(\lambda_t)$ and a global intervention strength $\alpha \geq 0$, this balance is expressed as

$$\tilde{c}_t = \arg\min_{u\in\mathbb{R}^k} \frac{1}{2}\|u - c_t\|_2^2 + \frac{\alpha}{2}\|u - a_t\|_{\Lambda_t}^2, \tag{8}$$

where $\|v\|_{\Lambda_t}^2 = v^\top \Lambda_t v$. The first term penalizes displacement from the current coordinates, while the second encourages proximity to the anchor according to the predicted direction weights. The parameter α controls their relative importance. This strictly convex quadratic objective admits a unique closed-form solution, requiring no iterative optimization:

$$\tilde{c}_t = c_t - \gamma_t \odot (c_t - a_t), \qquad \gamma_{t,i} = \frac{\alpha\lambda_{t,i}}{1 + \alpha\lambda_{t,i}}.$$

Here, $\odot$ denotes element-wise multiplication, and $\gamma_{t,i} \in [0, 1)$ specifies the fraction of the distance traveled toward the anchor in coordinate i. Each coordinate therefore moves toward its target without overshooting, with $\alpha = 0$ recovering the original coordinates. Reconstructing the hidden state gives

$$\begin{aligned}\tilde{h}_t &= z_t + B\tilde{c}_t\\ &= h_t - B[\gamma_t \odot (c_t - a_t)].\end{aligned} \tag{9}$$

The resulting change $\tilde{h}_t - h_t$ lies in $\mathrm{span}(B)$, preserving the orthogonal residual at the current token and layer. Appendix C provides the derivation and analysis of the update. At inference, SCAN applies this closed-form correction once at the selected layer before continuing the same backbone forward pass, requiring neither a contrastive branch nor an additional backbone forward pass.

3.3 Subspace Construction and Navigator Training

Hallucination-Associated Subspace Construction. SCAN uses task-specific response pairs to identify shared directions associated with object hallucination. For captioning and object existence tasks, the frozen LVLM generates N candidate responses per image–prompt input. Using dataset object annotations, we pair a grounded caption y_i^+ with the highest-hallucination caption y_i^- for the same input. For object existence, y_i^+ and y_i^- instead give correct and incorrect judgments about the queried object's presence, respectively.

Under teacher forcing, we extract each response's answer-position hidden states $h_{i,t}^{\pm}$ at the intervention layer. Averaging over its $T_i^{\pm}$ answer positions yields a response-level representation for computing the paired difference:

$$\begin{aligned} \bar{h}_i^{\pm} &= \frac{1}{T_i^{\pm}} \sum_{t=1}^{T_i^{\pm}} h_{i,t}^{\pm}, \\ \Delta_i &= \bar{h}_i^{-} - \bar{h}_i^{+}. \end{aligned} \tag{10}$$

To identify dominant directions across pairs, we stack the n differences into $D \in \mathbb{R}^{n \times d}$ and compute its singular value decomposition:

$$D = [\Delta_1, \ldots, \Delta_n]^\top = U \Sigma V^\top. \tag{11}$$

The leading k right singular vectors form $B = [v_1, \ldots, v_k]$, satisfying $B^\top B = I_k$ as required by Eq. (1). This basis remains frozen thereafter.

Preference-Based Navigator Training. SCAN trains the Navigator using response preferences to favor grounded descriptions and correct object judgments, with the LVLM backbone and B frozen. Let π_θ denote the resulting model and π_0 the unmodified reference. Under teacher forcing, each response receives an object-weighted, length-normalized score:

$$S_j(y \mid x) = \frac{\sum_t w_t \log \pi_j(y_t \mid x, y_{<t})}{\sum_t w_t}. \tag{12}$$

Here, $j \in \{\theta, 0\}$, x is the image–prompt input, and t ranges over answer positions $\mathcal{T}(y)$. For both models, $w_t = w_{\mathrm{obj}} > 1$ at positions aligned to object mentions and $w_t = 1$ elsewhere.

Both tasks use pairs $p = (x, y^+, y^-)$ of preferred and rejected responses: grounded versus hallucinated captions, or correct versus incorrect yes/no answers. A shared modified DPO-style objective (Rafailov et al., 2023) encourages improvement over the reference margin:

$$\begin{aligned} \delta_j(p) &= S_j(y^+ \mid x) - S_j(y^- \mid x), \\ \mathcal{L}_q(p) &= -\log \sigma(\beta_q[\delta_\theta(p) - \delta_0(p)]). \end{aligned} \tag{13}$$

Here, $q \in \{\mathrm{cap}, \mathrm{exist}\}$ identifies the task, $\beta_q > 0$ sets its preference scale, and σ is the logistic sigmoid.

To discourage large corrections, we penalize relative hidden-state displacement over $\mathcal{T}(p) = \mathcal{T}(y^+) \uplus \mathcal{T}(y^-)$, the disjoint union of answer positions in the two teacher-forced responses:

$$\mathcal{L}_{\mathrm{reg}}(p) = \frac{\eta_h}{|\mathcal{T}(p)|} \sum_{t \in \mathcal{T}(p)} \frac{\left\| \tilde{h}_t - h_t \right\|_2}{\max(\|h_t\|_2, \epsilon)}. \tag{14}$$

Here, h_t and $\tilde{h}_t$ are the states before and after correction, $\eta_h \geq 0$ controls regularization strength, and $\epsilon > 0$ stabilizes the denominator. The full objective combines both tasks and regularization:

$$\mathcal{L}_{\mathrm{nav}} = \sum_q \mathbb{E}_{p \sim \mathcal{D}_q}[\mathcal{L}_q(p)] + \mathbb{E}_{p \sim \mathcal{D}_{\mathrm{mix}}}[\mathcal{L}_{\mathrm{reg}}(p)]. \tag{15}$$

Here, $\mathcal{D}_q$ is the preference dataset for task q, and $\mathcal{D}_{\mathrm{mix}}$ is the distribution used to sample pairs from both tasks for regularization. Appendices D and F provide implementation and data details.

Table 1: POPE and CHAIR results from our runs on four LVLMs, except † entries from VTI. Bold marks the best results; "–" denotes unevaluated settings whose required hyperparameters are unspecified in the original papers.

Model	Method	Random		Popular		Adversarial		CHAIR	
		Acc.↑	F1↑	Acc.↑	F1↑	Acc.↑	F1↑	$C_s \downarrow$	$C_i \downarrow$
LLaVA-1.5 7B	Vanilla	85.1	83.7	83.7	82.4	80.9	80.0	48.0	14.3
	VCD	86.3	85.2	84.5	83.5	81.4	80.9	51.8	15.0
	ICD	84.4	82.2	83.8	81.6	81.5	79.5	50.6	14.6
	VTI	84.5	82.7	83.3	81.7	80.7	79.3	41.4	11.4
	ASCD	88.8	88.0	86.5	85.8	83.4	83.1	46.6	11.2
	SCAN	**89.2**	**88.6**	**87.1**	**86.7**	**83.8**	**83.7**	**31.6**	**10.6**
LLaVA-1.5 13B	Vanilla	85.4	84.2	84.4	83.3	82.0	81.2	48.8	13.1
	VCD	88.5	87.5	87.8	86.9	85.4	84.7	49.0	13.7
	ICD	88.0	86.7	87.4	86.2	85.5	84.4	49.6	13.1
	VTI	85.5	84.2	84.6	83.3	82.2	81.3	32.8	9.2
	ASCD	87.3	85.8	86.3	84.9	84.3	83.0	50.4	13.4
	SCAN	**89.7**	**88.9**	**88.2**	**87.5**	**85.7**	**85.2**	**31.8**	**9.0**
LLaVA-NeXT-7B	Vanilla	85.1	83.1	83.6	81.7	81.5	79.8	30.4	8.9
	VCD	87.7	86.4	86.4	85.1	85.3	84.0	35.6	10.0
	ICD	83.5	80.3	83.0	79.8	82.3	79.2	32.4	9.6
	VTI	85.7	83.8	84.1	82.3	81.9	80.4	28.2	8.3
	ASCD	86.4	84.4	85.7	83.7	84.8	82.8	36.0	10.0
	SCAN	**90.2**	**89.5**	**88.8**	**88.1**	**87.1**	**86.5**	**22.0**	**5.7**
Qwen-VL-Chat	Vanilla	84.5	82.3	83.4	81.4	80.5	78.9	38.6	11.9
	VCD	85.5	83.4	84.3	82.3	82.0	80.2	46.4	14.2
	ICD	85.4	83.6	84.1	82.4	81.7	80.3	46.8	13.0
	VTI†	86.7	85.6	85.7	84.5	83.1	82.2	–	–
	ASCD	–	–	–	–	–	–	–	–
	SCAN	**88.4**	**87.6**	**86.6**	**86.0**	**83.3**	**83.1**	**24.4**	**7.5**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Benchmarks and Metrics. Our evaluation covers POPE (Li et al., 2023b), CHAIR (Rohrbach et al., 2018), and the AMBER generative set (Wang et al., 2023). On POPE, we report accuracy and F1 across the Random, Popular, and Adversarial splits. On CHAIR, we report sentence- and instance-level hallucination rates (C_s and C_i), together with object recall. For the AMBER generative set, we report CHAIR, Cover, Hal, and Cog.

Models and Baselines. We evaluate six LVLMs, including LLaVA-1.5 7B and 13B (Liu et al., 2024b), LLaVA-NeXT-7B (Liu et al., 2024c), Qwen-VL-Chat (Bai et al., 2023), Qwen2.5-VL-7B-Instruct (Bai et al., 2025b), and Qwen3-VL-8B-Instruct (Bai et al., 2025a). We compare SCAN with VCD (Leng et al., 2024), ICD (Wang et al., 2024), OPERA (Huang et al., 2024), ASCD (Wang et al., 2026), VTI (Liu et al., 2025a), Nullu (Yang et al., 2025), and HulluEdit (Lin et al., 2026).

Implementation Details. We build a rank-128 basis from 3,072 response pairs and train the approximately 4M-parameter Navigator on 5,334 preference pairs in about 50 minutes on one NVIDIA RTX 4090. Inference uses $\alpha = 0.10$. CHAIR generation limits are 512 tokens in Table 1 and 64 in Tables 2 and 4. Appendices G and H detail hyperparameters, sensitivity analyses, and evaluation protocols.

4.2 OVERALL PERFORMANCE

Results on POPE. SCAN achieves the highest POPE accuracy and F1 across all four LVLMs and all three splits in Table 1. On LLaVA-1.5 7B's Adversarial split, SCAN reaches 83.8 accuracy and 83.7 F1, exceeding ASCD, the strongest baseline, by 0.4 and 0.6 percentage points, respectively.

Table 2: CHAIR results on LLaVA-1.5 7B.

Method	$C_s \downarrow$	$C_i \downarrow$	Recall↑
Vanilla	20.40	7.08	**71.1**
OPERA	19.00	6.25	68.2
Nullu	14.80	5.68	66.6
HulluEdit	13.00	4.18	56.2
SCAN	**10.00**	**4.02**	64.0

Table 3: AMBER results on LLaVA-1.5 7B.

Method	CHAIR↓	Cover↑	Hal↓	Cog↓
Vanilla	8.3	51.8	37.8	4.5
VCD	8.7	52.0	41.4	4.4
ICD	8.5	**52.2**	39.3	4.7
ASCD	6.5	51.2	32.2	2.9
SCAN	**5.6**	50.4	**23.9**	**1.6**

Table 4: POPE and CHAIR results on Qwen2.5-VL and Qwen3-VL.

Model	Method	Random		Popular		Adversarial		CHAIR	
		Acc.↑	F1↑	Acc.↑	F1↑	Acc.↑	F1↑	$C_s \downarrow$	$C_i \downarrow$
Qwen2.5-VL-7B	Vanilla	87.2	85.5	86.6	85.0	85.4	83.9	10.4	5.3
	SCAN	**90.4**	**89.6**	**89.2**	**88.5**	**86.8**	**86.2**	**10.2**	**4.7**
Qwen3-VL-8B	Vanilla	91.2	90.6	88.5	88.0	86.7	86.3	9.2	5.0
	SCAN	**91.4**	**90.7**	**88.8**	**88.3**	**87.0**	**86.7**	**6.4**	**3.5**

Results on CHAIR. Table 2 shows that SCAN achieves the lowest C_s and C_i under the 64-token limit, with scores of 10.00 and 4.02 compared with HulluEdit's 13.00 and 4.18, while achieving higher object recall. Under the 512-token limit in Table 1, SCAN achieves the lowest reported C_s and C_i across all four LVLMs. On LLaVA-1.5 7B, its C_s reaches 31.6, improving over VTI and Vanilla by 9.8 and 16.4 percentage points, respectively.

Results on AMBER. Table 3 shows that SCAN achieves the lowest CHAIR, Hal, and Cog scores on the AMBER generative set, reaching 5.6, 23.9, and 1.6 compared with ASCD's 6.5, 32.2, and 2.9. Cover decreases from 51.2 to 50.4, indicating a modest coverage cost.

Results on Recent LVLMs. Table 4 shows that the POPE and CHAIR improvements over Vanilla also extend to Qwen2.5-VL-7B and Qwen3-VL-8B. On Qwen2.5-VL-7B, Random-split accuracy rises from 87.2 to 90.4, while C_s falls from 10.4 to 10.2 and C_i from 5.3 to 4.7. On Qwen3-VL-8B, C_s falls from 9.2 to 6.4 and C_i from 5.0 to 3.5.

4.3 Ablation Studies

Table 5: CHAIR ablations on LLaVA-1.5 7B at a 64 token limit. Entries marked ‡ come from repetitive generation.

(a) Navigation Components

Variant	$C_s \downarrow$	$C_i \downarrow$
SCAN	**10.00**	**4.02**
Vanilla	20.40	7.08
Fixed + uniform	15.00	5.88
Fixed + direction-wise	12.80	4.68

(b) Subspace basis

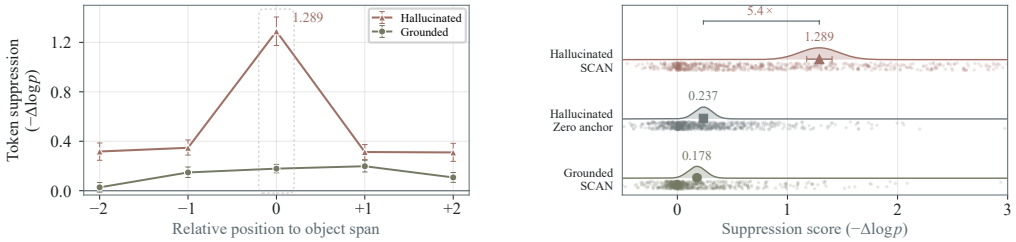
Variant	$C_s \downarrow$	$C_i \downarrow$
SCAN	**10.00**	**4.02**
Random orthogonal	19.80	6.49
Shuffled-pair SVD	13.80	6.54

(c) Navigator input

Variant	$C_s \downarrow$	$C_i \downarrow$
SCAN	**10.00**	**4.02**
Full-state, $\alpha = 0.02^{\ddagger}$	10.80	3.82
Full-state, $\alpha = 0.05^{\ddagger}$	6.00	2.44
Full-state, $\alpha = 0.10^{\ddagger}$	2.00	1.27

Navigation Components. With a fixed anchor, direction-wise weights reduce C_s from 15.00 to 12.80 relative to uniform strengths, at object recall around 65%. An adaptive anchor further reduces C_s to 10.00 and C_i from 4.68 to 4.02.

Subspace Basis. Paired-difference SVD outperforms random orthogonal and shuffled-pair bases, whose C_s values rise to 19.80 and 13.80, respectively. Both also increase C_i, supporting the use of aligned grounded–hallucinated response differences.



(a) Suppression around object spans.

(b) Object-span suppression distributions.

Figure 3: Object-span suppression on LLaVA-1.5 7B.

Navigator Input. Conditioning on the full hidden state h_t causes repetitive degeneration: the loop rate rises from 5.0% at $\alpha = 0.02$ to 93.6% at $\alpha = 0.10$. Repetition suppresses object mentions, so the low scores marked ‡ do not establish improved grounding. Residual input z_t instead yields a 0.0% loop rate in this evaluation, supporting it as the more stable choice. Appendix E analyzes residual conditioning and provides detailed repetition and object-coverage results.

4.4 Further Analysis

Mechanism Analysis. Figure 3 compares SCAN with a zero-anchor control that sets a_t to $\mathbf{0}_k$, with matched intervention strengths. The analysis uses 1,000 held-out grounded and hallucinated response pairs. The left panel aligns token profiles to the first object span, while the right panel compares SCAN with the zero-anchor control on hallucinated spans. SCAN reaches a mean suppression of 1.289 at hallucinated spans, about 5.4 times the 0.237 observed with the zero-anchor control. In contrast, its mean suppression at grounded spans remains 0.178, supporting selective suppression of hallucinated object mentions rather than uniform suppression of object mentions.

Efficiency Analysis. Table 6 compares batch-size-1 decoding on LLaVA-1.5 7B after 20 warm-up runs; speeds are relative to Vanilla. SCAN retains 98.4% throughput at 48.21 tokens/s with 0.02 GB more memory. VCD and HulluEdit achieve 24.67 and 21.55 tokens/s. HulluEdit uses less memory, but SCAN is substantially faster, consistent with a lightweight Navigator and single-pass decoding.

Table 6: Decoding efficiency.

Method	Tokens/s↑	Speed (%)↑	Mem. (GB)↓
Vanilla	48.98 ± 0.24	100.0	14.45
SCAN	48.21 ± 0.04	98.4	14.47
VCD	24.67 ± 0.03	50.4	15.15
HulluEdit	21.55 ± 0.01	44.0	14.04

Qualitative Analysis. Figure 4 compares responses to the same image and prompt. Vanilla describes a "pizza box" on the bicycle's rear rack. SCAN instead identifies "cakes" and retains the grounded "green bicycle" detail. Rather than avoiding the object altogether, it corrects the unsupported identity while preserving the surrounding visual description. This example provides a qualitative counterpart to the aggregate hallucination reductions.

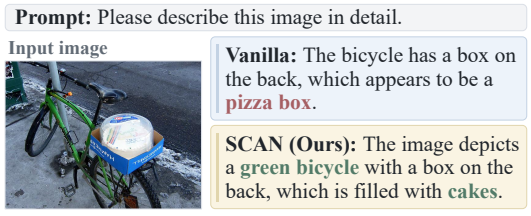


Figure 4: Qualitative comparison.

5 Conclusion

In this paper, we identify fixed-target mismatch as a key limitation of subspace interventions that use the same input-independent target across contexts, potentially suppressing grounded object mentions and exacerbating the hallucination–recall trade-off. To address this limitation, we propose SCAN, which retains a shared hallucination-associated subspace but uses the orthogonal residual to predict a context-adaptive anchor and direction-wise navigation strengths for each input. Experiments show that SCAN consistently reduces object hallucinations, improves the hallucination–recall trade-off over fixed-target baselines, and retains near-baseline decoding efficiency.

AI USE STATEMENT

Generative AI assisted conceptual framing of the fixed-target mismatch analysis, mathematical formulation and proof drafting/checking, assistance with implementing experimental code, feedback on research methodology and experimental design, feedback on the presentation of methodological explanations and experimental comparisons, and interpretation of author-supplied results. It also assisted literature discovery and bibliography preparation, as well as language editing. We have not used generative AI tools to generate synthetic datasets or to clean or reformat datasets. These tools generated no experimental measurements or benchmark scores. The authors are responsible for the scientific conclusions, attribution, and final submission.

REPRODUCIBILITY STATEMENT

We provide implementation and evaluation details to facilitate reproducibility. The main algorithm and training procedure are described in the main text, while Appendix G provides the hyperparameters and implementation details, and Appendix H specifies the evaluation protocols. Additional ablations and sensitivity analyses are also reported in the appendix.

REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025b.

Zhiyuan Chen, Yuecong Min, Jie Zhang, Bei Yan, Jiahao Wang, Xiaozhen Wang, and Shiguang Shan. A survey of multimodal hallucination evaluation and detection. *International Journal of Computer Vision*, 134:131, 2026.

Hamidreza Dastmalchi, Aijun An, Ali Cheraghian, and Hamed Barzamini. Fighting hallucinations with counterfactuals: Diffusion-guided perturbations for LVLM hallucination suppression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4178–4187, 2026.

Geert Heyman and Frederik Vandeputte. Steer like the LLM: Activation steering that mimics prompting. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2026.

Ailin Huang, Chengyuan Yao, Chunrui Han, et al. STEP3-VL-10B technical report, 2026.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. OPERA: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13418–13427, 2024.

Chaoya Jiang, Haiyang Xu, Mengfan Dong, Jiaxing Chen, Wei Ye, Ming Yan, Qinghao Ye, Ji Zhang, Fei Huang, and Shikun Zhang. Hallucination augmented contrastive learning for multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27036–27046, 2024.

Prannay Kaul, Zhizhong Li, Hao Yang, Yonatan Dukler, Ashwin Swaminathan, C. J. Taylor, and Stefano Soatto. THRONE: An object-based hallucination benchmark for the free-form generations of large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27228–27238, 2024.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13872–13882, 2024.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 41451–41530, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 292–305, 2023b.

Yangguang Lin, Quan Fang, Yufei Li, Jiachen Sun, Junyu Gao, and Jitao Sang. HulluEdit: Single-pass evidence-consistent subspace editing for mitigating hallucinations in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11086–11095, 2026.

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26296–26306, 2024b.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. LLaVA project blog, January 30, 2024c.

Sheng Liu, Haotian Ye, and James Y. Zou. Reducing hallucinations in large vision-language models via latent space steering. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025a.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Haotian Tang, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Jinyi Hu, Sifei Liu, Ranjay Krishna, Pavlo Molchanov, Jan Kautz, Hongxu Yin, Song Han, and Yao Lu. NVILA: Efficient frontier visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4122–4134, 2025b.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 53728–53741, 2023.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4035–4045, 2018.

Jingran Su, Jingfan Chen, Hongxin Li, Yuntao Chen, Li Qing, and Zhaoxiang Zhang. Activation steering decoding: Mitigating hallucination in large vision-language models through bidirectional hidden state intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12964–12974, 2025.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and Jitao Sang. AMBER: An LLM-free multi-dimensional benchmark for MLLMs hallucination evaluation, 2023.

Xintong Wang, Jingheng Pan, Liang Ding, and Chris Biemann. Mitigating hallucinations in large vision-language models with instruction contrastive decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 15840–15853, 2024.

Yujun Wang, Aniri, Jinhe Bi, Soeren Pirk, and Yunpu Ma. ASCD: Attention-steerable contrastive decoding for reducing hallucination in MLLM. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 40, pp. 10306–10314, 2026.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 63908–63962, 2024.

Le Yang, Ziwei Zheng, Boxu Chen, Zhengyu Zhao, Chenhao Lin, and Chao Shen. Nullu: Mitigating object hallucinations in large vision-language models via HalluSpace projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14635–14645, 2025.

Manjiang Yu, Hongji Li, Junwei Chen, Xue Li, Priyanka Singh, Yang Cao, and Lijie Hu. Multi-adapter representation interventions via energy calibration. In *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2026.

Hongjue Zhao, Haosen Sun, Jiangtao Kong, Xiaochang Li, Qineng Wang, Liwei Jiang, Qi Zhu, Tarek Abdelzaher, Yejin Choi, Manling Li, and Huajie Shao. ODESteer: A unified ODE-based steering framework for LLM alignment. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

Zhiyuan Zhao, Bin Wang, Linke Ouyang, Xiaoyi Dong, Jiaqi Wang, and Conghui He. Beyond multimodal hallucinations: Enhancing LVLMs through hallucination-aware direct preference optimization. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2025.

APPENDIX CONTENTS

A Notation . . . 14

B Fixed-Target Mismatch Analysis . . . 14

B.1 Formal Setup . . . 14

B.2 Conditional Optimality . . . 15

B.3 Best Fixed Target and Irreducible Contextual Variation . . . 15

B.4 Criterion for a Learned Context-Dependent Rule . . . 16

B.5 Effect of Conditioning Information . . . 16

C SCAN Navigation Derivations . . . 17

C.1 Conditional Proximal Navigation Update . . . 17

C.2 Gain-Weighted Target Mismatch . . . 18

C.3 Structural Properties of the Update . . . 19

D Algorithms and Navigator Architecture . . . 20

D.1 Hallucination-Associated Basis Construction . . . 20

D.2 Navigator Training . . . 21

D.3 Autoregressive Inference . . . 22

D.4 Navigator Architecture and Complexity . . . 23

E Residual Conditioning and Generation Stability . . . 24

E.1 Local Properties of Residual Conditioning . . . 24

E.2 Intervention Effects Across Decoding Steps . . . 24

E.3 Controlled Comparison of Navigator Inputs . . . 25

E.4 Empirical Analysis of Generation Stability . . . 25

F Data Construction and Preprocessing . . . 26

F.1 Data Sources and Splits . . . 26

F.2 Pair Construction and Token Weighting . . . 27

G Hyperparameters and Sensitivity Analysis . . . 27

G.1 Default Configuration . . . 27

G.2 Sensitivity Analysis . . . 28

H Evaluation Protocol . . . 29

H.1 Prompts and Model Formatting . . . 29

H.2 Decoding and Metric Computation . . . 29

A NOTATION

Table 7 summarizes the main symbols used throughout the appendix. The index t denotes the current decoding position, whereas Z and $C^\dagger$ are random variables in the population analysis. The desired anchor $C^\dagger$ is used only for analysis, not as a directly observed training label.

Table 7: Main notation used throughout the appendix.

Symbol	Meaning
Subspace decomposition	
h_t	Hidden state at the intervention layer, in $\mathbb{R}^d$
B	Fixed subspace basis in $\mathbb{R}^{d\times k}$, with $B^\top B = I_k$
c_t	Subspace coordinates $B^\top h_t \in \mathbb{R}^k$
z_t	Orthogonal residual $h_t - Bc_t \in \mathbb{R}^d$
Anchor prediction and navigation	
a_t	Predicted anchor in $\mathbb{R}^k$
ρ_t	Unconstrained Navigator output for direction weights, in $\mathbb{R}^k$
λ_t	Positive direction weights $\mathrm{softplus}(\rho_t)$
α	Global intervention strength, $\alpha \geq 0$
γ_t	Direction-wise gains, $\gamma_{t,i} = \alpha\lambda_{t,i}/(1+\alpha\lambda_{t,i})$
$\tilde{c}_t$	Updated coordinates $c_t - \gamma_t \odot (c_t - a_t)$
$\tilde{h}_t$	Reconstructed hidden state $z_t + B\tilde{c}_t$
Target approximation analysis	
Z	Random residual context
$C^\dagger$	Desired anchor in $\mathbb{R}^k$, used only for analysis
$\mu(Z)$	Conditional mean target $\mathbb{E}[C^\dagger \mid Z]$
b	Fixed target in $\mathbb{R}^k$ shared across contexts
$\bar{\mu}$	Mean target $\mathbb{E}[\mu(Z)]$
$\mathcal{R}(a)$	Squared-error risk $\mathbb{E}[\|C^\dagger - a(Z)\|_2^2]$ of a target rule $a(Z)$
$\mathcal{M}(b)$	Fixed-target mismatch $\mathcal{R}(b) - \mathcal{R}(\mu)$
$\mathcal{M}_{\mathrm{fix}}$	Minimum fixed-target mismatch $\min_{b\in\mathbb{R}^k} \mathcal{M}(b)$
V_{ctx}	Contextual target variation $\mathbb{E}[\|\mu(Z) - \bar{\mu}\|_2^2]$
$\Gamma(Z)$	Prescribed Z-measurable diagonal gain rule, held fixed across target rules

B FIXED-TARGET MISMATCH ANALYSIS

B.1 FORMAL SETUP

To isolate target selection, we fix an intervention layer and an orthonormal basis $B \in \mathbb{R}^{d\times k}$ with $B^\top B = I_k$, and abstract away intervention strength and the update rule. Let Z denote the residual context available for target selection, and let $C^\dagger \in \mathbb{R}^k$ denote the desired anchor in subspace coordinates. All target rules are evaluated under the same joint distribution of $(Z, C^\dagger)$, induced by a fixed sampling procedure over multimodal inputs, response prefixes, and decoding positions.

The variable $C^\dagger$ specifies the desired destination for the sampled context in this approximation problem. It is used only for analysis and is not a directly observed training label. We assume $\mathbb{E}[\|C^\dagger\|_2^2] < \infty$.

An admissible target rule is a measurable function a satisfying $\mathbb{E}[\|a(Z)\|_2^2] < \infty$. Define

$$\mathcal{R}(a) = \mathbb{E}\left[\|C^\dagger - a(Z)\|_2^2\right], \qquad \mu(Z) = \mathbb{E}[C^\dagger \mid Z]. \tag{16}$$

By conditional Jensen's inequality,

$$\mathbb{E}[\|\mu(Z)\|_2^2] \leq \mathbb{E}[\|C^\dagger\|_2^2] < \infty,$$

so μ is itself admissible. The risk $\mathcal{R}$ measures target approximation; it does not specify a quantitative relation between anchor distance and object hallucination.

B.2 Conditional Optimality

We first identify the optimal target rule based on the residual. This provides a common reference for comparing fixed and context-dependent targets.

Lemma 2 (Conditional optimality)**.** *Every admissible rule a satisfies*

$$\mathcal{R}(a) = \mathcal{R}(\mu) + \mathbb{E}\Big[\|a(Z) - \mu(Z)\|_2^2\Big]. \tag{17}$$

Consequently, μ uniquely minimizes $\mathcal{R}$ among admissible rules, up to almost-sure equality.

Proof. Decompose the prediction error as

$$C^\dagger - a(Z) = \big(C^\dagger - \mu(Z)\big) + \big(\mu(Z) - a(Z)\big).$$

The second term is measurable with respect to Z, while

$$\mathbb{E}[C^\dagger - \mu(Z) \mid Z] = 0. \tag{18}$$

The cross term therefore has zero expectation. Expanding the squared norm and taking expectations gives

$$\mathcal{R}(a) - \mathcal{R}(\mu) = \mathbb{E}\Big[\|a(Z) - \mu(Z)\|_2^2\Big] \geq 0.$$

Equality holds if and only if $a(Z) = \mu(Z)$ almost surely. □

This is the orthogonal-projection property of conditional expectation under squared loss. It expresses the excess risk of any target rule through its discrepancy from $\mu(Z)$. For a constant rule, this discrepancy admits the further decomposition below.

B.3 Best Fixed Target and Irreducible Contextual Variation

To characterize the restriction imposed by a fixed target, let $\bar{\mu} = \mathbb{E}[\mu(Z)] = \mathbb{E}[C^\dagger]$ and define

$$V_{\mathrm{ctx}} = \mathbb{E}\Big[\|\mu(Z) - \bar{\mu}\|_2^2\Big] = \mathrm{Tr}(\mathrm{Cov}[\mu(Z)]). \tag{19}$$

For a fixed target $b \in \mathbb{R}^k$, its mismatch is

$$\mathcal{M}(b) := \mathbb{E}\Big[\|\mu(Z) - b\|_2^2\Big] = \mathcal{R}(b) - \mathcal{R}(\mu),$$

where the equality follows from Lemma 2.

Proposition 2 (Fixed-target approximation gap)**.** *Every fixed target $b \in \mathbb{R}^k$ satisfies*

$$\mathcal{M}(b) = V_{\mathrm{ctx}} + \|b - \bar{\mu}\|_2^2. \tag{20}$$

The unique best fixed target is $b^\star = \bar{\mu}$, and

$$\mathcal{M}_{\mathrm{fix}} := \min_{b \in \mathbb{R}^k} \mathcal{M}(b) = V_{\mathrm{ctx}}.$$

This minimum mismatch is strictly positive if and only if $\mu(Z)$ is not almost surely constant.

Proof. Expanding $\mu(Z) - b$ around $\bar{\mu}$ gives

$$\begin{aligned}\mathcal{M}(b) = \mathbb{E}\Big[\|\mu(Z) - \bar{\mu}\|_2^2\Big] + \|b - \bar{\mu}\|_2^2 \\ + 2\mathbb{E}\big[(\mu(Z) - \bar{\mu})^\top (\bar{\mu} - b)\big].\end{aligned} \tag{21}$$

Since $\mathbb{E}[\mu(Z) - \bar{\mu}] = 0$, the cross term vanishes, yielding Eq. (20). The second term is uniquely minimized at $b = \bar{\mu}$. Moreover, $V_{\mathrm{ctx}} = 0$ if and only if $\mu(Z) = \bar{\mu}$ almost surely, which establishes the strict-positivity condition. □

Equation (20) separates the effect of choosing a particular fixed target from the restriction that the target must remain constant. Optimizing b removes the offset $\|b - \bar{\mu}\|_2^2$, but leaves the contextual variation V_{ctx} unchanged. Thus, the relevant source of irreducible mismatch is variation in the conditional mean $\mu(Z)$, rather than arbitrary variation in $C^\dagger$. If $\mu(Z)$ is constant almost surely, a fixed target already attains the optimal risk based on Z.

Corollary 3 (Zero-target specialization)**.** *The zero target is the best fixed target if and only if $\bar{\mu} = \mathbf{0}_k$. Its mismatch is*

$$\mathcal{M}(\mathbf{0}_k) = V_{\mathrm{ctx}} + \|\bar{\mu}\|_2^2 . \tag{22}$$

Relative to the best fixed target, its additional excess risk is

$$\mathcal{R}(\mathbf{0}_k) - \min_{b \in \mathbb{R}^k} \mathcal{R}(b) = \|\bar{\mu}\|_2^2 . \tag{23}$$

These identities follow by substituting $b = \mathbf{0}_k$ into Eq. (20). Even when the zero target is optimal among fixed targets, it retains the gap V_{ctx} whenever $\mu(Z)$ is nonconstant. The same gap also follows from the vector law of total variance.

B.4 Criterion for a Learned Context-Dependent Rule

The preceding result compares fixed targets with the optimal conditional rule μ. A learned anchor generally approximates this rule, so its approximation error determines whether it improves on the best fixed target.

Corollary 4 (Context-dependent approximation criterion)**.** *For every admissible rule $\widehat{a}(Z)$,*

$$\mathcal{R}(\widehat{a}) - \min_{b \in \mathbb{R}^k} \mathcal{R}(b) = \mathbb{E}\Big[\|\widehat{a}(Z) - \mu(Z)\|_2^2\Big] - V_{\mathrm{ctx}}. \tag{24}$$

Thus, $\widehat{a}$ has strictly lower squared-error risk than every fixed target if and only if its approximation error relative to μ is smaller than V_{ctx}.

Proof. Combining Lemma 2 with Proposition 2 gives

$$\begin{aligned} \mathcal{R}(\widehat{a}) - \min_{b \in \mathbb{R}^k} \mathcal{R}(b) &= \mathcal{R}(\widehat{a}) - \mathcal{R}(\mu) - V_{\mathrm{ctx}} \\ &= \mathbb{E}\Big[\|\widehat{a}(Z) - \mu(Z)\|_2^2\Big] - V_{\mathrm{ctx}}. \end{aligned}$$

□

The criterion identifies when a learned anchor can exploit contextual target variation. In SCAN, the Navigator is trained using response preferences rather than direct supervision of $C^\dagger$. The analysis therefore motivates learning an adaptive target, while the effects of the learned intervention on hallucination and recall are evaluated empirically.

B.5 Effect of Conditioning Information

The optimal risk also depends on the information available to the target rule. We now characterize how refining this information affects target approximation and apply the result to residual conditioning.

For a sub-σ-algebra $\mathcal{G}$ representing the available information, define

$$\mu_{\mathcal{G}} = \mathbb{E}[C^\dagger \mid \mathcal{G}], \qquad \mathcal{R}^\star_{\mathcal{G}} = \mathbb{E}\Big[\big\|C^\dagger - \mu_{\mathcal{G}}\big\|_2^2\Big] .$$

By the same conditional-expectation argument as in Lemma 2, $\mathcal{R}^\star_{\mathcal{G}}$ is the minimum risk among square-integrable targets measurable with respect to $\mathcal{G}$.

Proposition 3 (Information-refinement monotonicity)**.** *If $\mathcal{G}_1 \subseteq \mathcal{G}_2$, then*

$$\mathcal{R}^\star_{\mathcal{G}_1} - \mathcal{R}^\star_{\mathcal{G}_2} = \mathbb{E}\Big[\|\mu_{\mathcal{G}_2} - \mu_{\mathcal{G}_1}\|_2^2\Big] \geq 0. \tag{25}$$

Thus, refining the available information cannot increase the optimal squared-error risk.

Proof. Write

$$C^\dagger - \mu_{\mathcal{G}_1} = \big(C^\dagger - \mu_{\mathcal{G}_2}\big) + \big(\mu_{\mathcal{G}_2} - \mu_{\mathcal{G}_1}\big).$$

Because $\mathcal{G}_1 \subseteq \mathcal{G}_2$, the second term is measurable with respect to $\mathcal{G}_2$. The first term has zero conditional expectation given $\mathcal{G}_2$. Expanding the squared norm and taking expectations therefore gives

$$\mathcal{R}^\star_{\mathcal{G}_1} = \mathcal{R}^\star_{\mathcal{G}_2} + \mathbb{E}\Big[\|\mu_{\mathcal{G}_2} - \mu_{\mathcal{G}_1}\|_2^2\Big],$$

which proves the claim. □

Let X denote a fuller context from which Z is a measurable function. Then $\sigma(Z) \subseteq \sigma(X)$, and Proposition 3 yields

$$\mathcal{R}^\star_{\sigma(Z)} - \mathcal{R}^\star_{\sigma(X)} = \mathbb{E}\Big[\big\|\mathbb{E}[C^\dagger \mid X] - \mathbb{E}[C^\dagger \mid Z]\big\|_2^2\Big]. \tag{26}$$

This difference is the optimal squared-error cost of restricting target prediction to the residual. It vanishes if and only if

$$\mathbb{E}[C^\dagger \mid X] = \mathbb{E}[C^\dagger \mid Z] \quad \text{almost surely.}$$

This condition characterizes sufficiency for conditional-mean prediction under squared loss; it does not require equality of the full conditional distributions.

SCAN does not assume that this equality holds. Residual conditioning instead restricts how the predicted anchor and weights depend on the editable coordinates. Its structural motivation and observed decoding stability are examined in Appendix E. Those empirical results concern trained predictors during generation, whereas the information-refinement result concerns optimal predictors under a common fixed distribution.

The results above characterize target approximation. Appendix C next examines how target discrepancies affect the subspace update when the navigation gains are held fixed.

C SCAN Navigation Derivations

C.1 Conditional Proximal Navigation Update

At each decoding position, the Navigator predicts an anchor and direction weights from the current residual. These predictions are held fixed while solving the proximal problem, which balances movement toward the anchor against displacement from the current coordinates.

Proposition 4 (Conditional proximal navigation update)**.** *Let $B^\top B = I_k$, $h_t = Bc_t + z_t$, and $B^\top z_t = 0$. Given $(a_t, \rho_t) = g_\theta(z_t)$, $\lambda_t = \mathrm{softplus}(\rho_t)$, and $\Lambda_t = \mathrm{diag}(\lambda_t)$, the proximal objective*

$$\begin{aligned}\tilde{c}_t = \arg\min_{u \in \mathbb{R}^k} \; & \frac{1}{2}\,\|u - c_t\|_2^2 \\ & + \frac{\alpha}{2}(u - a_t)^\top \Lambda_t (u - a_t)\end{aligned} \tag{27}$$

has a unique minimizer for every $\alpha \geq 0$. Define the direction-wise gains by

$$\gamma_{t,i} = \frac{\alpha\lambda_{t,i}}{1 + \alpha\lambda_{t,i}}.$$

The updated coordinates and reconstructed hidden state $\tilde{h}_t = z_t + B\tilde{c}_t$ satisfy

$$\begin{aligned}\tilde{c}_t &= c_t - \gamma_t \odot (c_t - a_t), \\ \tilde{h}_t &= h_t - B[\gamma_t \odot (c_t - a_t)], \\ (I_d - BB^\top)\tilde{h}_t &= z_t.\end{aligned} \tag{28}$$

Each updated coordinate lies between its current value and the corresponding anchor coordinate.

Proof. Since $\lambda_{t,i} > 0$, the Hessian $I_k + \alpha\Lambda_t$ is positive definite for every $\alpha \geq 0$. The objective therefore has a unique minimizer. Setting its gradient to zero gives

$$(I_k + \alpha\Lambda_t)\tilde{c}_t = c_t + \alpha\Lambda_t a_t. \tag{29}$$

Because Λ_t is diagonal, the solution is

$$\begin{aligned}\tilde{c}_{t,i} &= \frac{c_{t,i} + \alpha\lambda_{t,i}a_{t,i}}{1 + \alpha\lambda_{t,i}} \\ &= (1 - \gamma_{t,i})c_{t,i} + \gamma_{t,i}a_{t,i}.\end{aligned} \tag{30}$$

For finite $\alpha \geq 0$, we have $\gamma_{t,i} \in [0, 1)$, so each coordinate is a convex combination of its current value and its anchor value.

Substituting this solution into $\tilde{h}_t = z_t + B\tilde{c}_t$ yields the hidden-state update in Eq. (28). Its displacement lies in $\mathrm{span}(B)$, hence

$$(I_d - BB^\top)\tilde{h}_t = (I_d - BB^\top)h_t = z_t. \qquad \square$$

The predicted weights λ_t determine the relative attraction toward the anchor in each direction, while α controls its global strength. Together they determine γ_t, the fractions of the coordinate-wise distances traveled toward the anchor. The solution is evaluated directly, without iterative optimization.

C.2 Gain-Weighted Target Mismatch

Appendix B compares target rules through their approximation errors. We now examine how these errors affect the update when the navigation gains are held fixed.

Retain the joint distribution of $(Z, C^\dagger)$ and the admissibility assumptions from that analysis, with $\mu(Z) = \mathbb{E}[C^\dagger \mid Z]$. Let $C \in \mathbb{R}^k$ denote the current editable coordinates, and let $\Gamma(Z)$ be a prescribed, Z-measurable diagonal gain rule with entries in $[0, 1)$. All comparisons use the same current coordinates, gain rule, and joint distribution. For an admissible target rule $a(Z)$, define

$$\widetilde{C}_a = C - \Gamma(Z)(C - a(Z)).$$

Let $\widetilde{C}_\mu$ and $\widetilde{C}_\dagger$ denote the corresponding updates obtained by replacing $a(Z)$ with $\mu(Z)$ and $C^\dagger$, respectively. The gain-weighted target risk is

$$\begin{aligned}\mathcal{R}_\Gamma(a) &:= \mathbb{E}\Big[\big\|\Gamma(Z)(C^\dagger - a(Z))\big\|_2^2\Big] \\ &= \mathbb{E}\left[\left\|\widetilde{C}_a - \widetilde{C}_\dagger\right\|_2^2\right].\end{aligned}$$

Thus, this risk measures discrepancy from the update induced by the analysis-only desired anchor under the same gains.

Proposition 5 (Gain-weighted target discrepancy)**.** *Let $W(Z) = \Gamma(Z)^2$. For every admissible rule a,*

$$\mathbb{E}\left[\left\|\widetilde{C}_a - \widetilde{C}_\mu\right\|_2^2\right] = \mathbb{E}\Big[\|\Gamma(Z)(a(Z) - \mu(Z))\|_2^2\Big]. \tag{31}$$

Moreover,

$$\begin{aligned}\mathcal{R}_\Gamma(a) = \mathcal{R}_\Gamma(\mu) \\ + \mathbb{E}\Big[\|\Gamma(Z)(a(Z) - \mu(Z))\|_2^2\Big].\end{aligned} \tag{32}$$

Consequently, μ minimizes $\mathcal{R}_\Gamma$. The minimizer need not be unique when gains can be zero.

Proof. Subtracting the two updates gives

$$\widetilde{C}_a - \widetilde{C}_\mu = \Gamma(Z)(a(Z) - \mu(Z)),$$

which proves Eq. (31).

For the risk decomposition, write

$$C^\dagger - a(Z) = \big(C^\dagger - \mu(Z)\big) + \big(\mu(Z) - a(Z)\big).$$

Since $W(Z)$ and $\mu(Z) - a(Z)$ are measurable with respect to Z, the cross term in the weighted squared norm has zero expectation:

$$\begin{aligned}&\mathbb{E}\big[(C^\dagger - \mu(Z))^\top W(Z)(\mu(Z) - a(Z))\big] \\ &\quad = \mathbb{E}\big[\mathbb{E}[C^\dagger - \mu(Z) \mid Z]^\top W(Z)(\mu(Z) - a(Z))\big] = 0.\end{aligned}$$

Expanding the weighted norm therefore gives Eq. (32). The excess risk is nonnegative and vanishes precisely when $\Gamma(Z)(a(Z) - \mu(Z)) = 0$ almost surely. □

These identities connect target approximation to update discrepancy. Smaller gains attenuate a given anchor error, and zero gains remove its effect in the corresponding directions. The comparison concerns the analyzed state update, rather than a guarantee on downstream generation.

For a fixed target $b \in \mathbb{R}^k$, define

$$\mathcal{J}_\Gamma(b) = \mathbb{E}\big[(b - \mu(Z))^\top W(Z)(b - \mu(Z))\big]. \tag{33}$$

By Eq. (32), $\mathcal{J}_\Gamma(b) = \mathcal{R}_\Gamma(b) - \mathcal{R}_\Gamma(\mu)$. Let $A = \mathbb{E}[W(Z)]$ and $q = \mathbb{E}[W(Z)\mu(Z)]$. Expanding the quadratic gives

$$\mathcal{J}_\Gamma(b) = b^\top A b - 2b^\top q + \mathbb{E}[\mu(Z)^\top W(Z)\mu(Z)].$$

If A is positive definite, the unique best fixed target is

$$b^\star_\Gamma = A^{-1}q. \tag{34}$$

Completing the square yields

$$\begin{aligned}\mathcal{J}_\Gamma(b) = \mathcal{J}_\Gamma(b^\star_\Gamma) \\ + (b - b^\star_\Gamma)^\top A(b - b^\star_\Gamma),\end{aligned} \tag{35}$$

where

$$\mathcal{J}_\Gamma(b^\star_\Gamma) = \mathbb{E}[\mu(Z)^\top W(Z)\mu(Z)] - q^\top A^{-1}q.$$

This decomposition is the gain-weighted counterpart of the fixed-target mismatch decomposition in Appendix B. Optimizing the constant target removes its offset from $b^\star_\Gamma$, leaving the minimum matched-gain discrepancy. Because gains can vary with context, the optimal fixed target is generally a weighted mean and need not equal $\bar{\mu}$.

If $\Gamma(Z)$ is a constant positive-definite diagonal matrix Γ, then $b^\star_\Gamma = \bar{\mu}$ and

$$\mathcal{J}_\Gamma(b^\star_\Gamma) = \mathrm{Tr}\big(\Gamma^2 \,\mathrm{Cov}[\mu(Z)]\big).$$

When zero gains are allowed, they can mask contextual variation. Nonconstant $\mu(Z)$ alone therefore does not ensure a strictly positive matched-gain discrepancy.

Remark 1 (Matched-gain comparison). The comparison holds the entire gain rule $\Gamma(Z)$ fixed across target rules. Using the same global strength α is insufficient if the predicted direction weights differ. The results isolate target selection under shared gains; they do not compare jointly retrained anchor and gain predictors.

C.3 Structural Properties of the Update

The closed-form solution also characterizes the geometry, magnitude, and limiting behavior of a single intervention.

Corollary 5 (Structural properties)**.** *Under the conditions of Proposition 4, the SCAN update has the following properties:*

1. ***Subspace confinement.*** *The displacement satisfies* $\tilde{h}_t - h_t \in \mathrm{span}(B)$*, and* $(I_d - BB^\top)\tilde{h}_t = z_t$.

2. ***Coordinate-wise interpolation.*** *Each updated coordinate lies between its current value and its anchor value, with*

$$\tilde{c}_{t,i} - a_{t,i} = \frac{c_{t,i} - a_{t,i}}{1 + \alpha\lambda_{t,i}}.$$

3. ***Displacement magnitude.*** *The hidden-state displacement satisfies*

$$\begin{aligned} \left\| \tilde{h}_t - h_t \right\|_2 &= \left\| \gamma_t \odot (c_t - a_t) \right\|_2 \\ &\leq \left\| c_t - a_t \right\|_2 . \end{aligned} \tag{36}$$

4. ***Zero intervention.*** *Setting $\alpha = 0$ gives $\gamma_t = \mathbf{0}_k$ and recovers $\tilde{h}_t = h_t$.*

5. ***Large-strength limit.*** *Holding h_t, B, and the Navigator outputs fixed, as $\alpha \to \infty$, $\gamma_{t,i} \to 1$ and $\tilde{c}_{t,i} \to a_{t,i}$. Every finite $\alpha \geq 0$ retains $\gamma_{t,i} < 1$.*

Proof. Subspace confinement and coordinate-wise interpolation follow from Eqs. (28) and (30).

For the displacement magnitude, $B^\top B = I_k$ implies $\|Bv\|_2 = \|v\|_2$ for every $v \in \mathbb{R}^k$. Applying this identity to $v = \gamma_t \odot (c_t - a_t)$ gives the equality in Eq. (36). The inequality follows from $0 \leq \gamma_{t,i} < 1$.

Finally, $\alpha = 0$ gives $\gamma_t = \mathbf{0}_k$. With the current state and Navigator outputs fixed, $\lambda_{t,i} > 0$ implies

$$\lim_{\alpha \to \infty} \gamma_{t,i} = 1, \qquad \lim_{\alpha \to \infty} \frac{c_{t,i} - a_{t,i}}{1 + \alpha\lambda_{t,i}} = 0,$$

which proves convergence of each updated coordinate to its fixed anchor coordinate. □

Bounded gains control the fraction of the distance traveled toward the anchor. The actual displacement also depends on $\|c_t - a_t\|_2$, so an anchor far from the current coordinates can still induce a large correction. This explains the role of the relative displacement penalty used during Navigator training.

All properties above concern the current token and intervention layer. In autoregressive generation, earlier edits can change later states and predicted anchors. Coordinate-wise interpolation and local residual preservation therefore do not by themselves guarantee sequence-level stability.

D Algorithms and Navigator Architecture

This section presents the model-level construction, training, and inference procedures. For a response prefix u, $f_{\leq \ell}(x, u)$ denotes the hidden state at the final position of the image–prompt input x followed by u, at the output of intervention layer ℓ. The notation $f_{>\ell}$ denotes the remaining model layers and output head. In the training and inference algorithms, forward computations use the KV cache accumulated during the preceding prediction steps, including the effects of earlier SCAN interventions. Cache arguments are omitted for readability. Appendix G reports the hyperparameter settings and sensitivity studies.

D.1 Hallucination-Associated Basis Construction

The basis is constructed once for each backbone and then frozen. Algorithm 1 uses n for the number of valid paired differences retained after filtering. The candidate source $\mathcal{G}$ may combine multiple generation configurations as described in Appendix F. For each input, y_i^+ and y_i^- denote the preferred and rejected responses selected according to the task-specific pairing rule. For $s \in \{+, -\}$, T_i^s is the number of answer tokens in candidate y_i^s. Using the frozen backbone without SCAN interventions, we extract the raw transformer-block output at each answer token's own position under teacher forcing: $h_{i,t}^s = f_{\leq \ell}(x_i, y_{i,\leq t}^s)$ for $t = 1, \ldots, T_i^s$.

For both tasks, we average the answer-token states within each response and subtract the preferred response representation from the rejected one. The resulting paired differences are stacked into D without subtracting their across-pair mean. SVD therefore captures the empirical second moment

Algorithm 1 Hallucination-Associated Basis

Input Frozen LVLM f, layer ℓ, rank k, image–prompt set $\mathcal{D}_{\text{basis}}$ with task labels and object annotations, candidate source $\mathcal{G}$
Output Orthonormal basis $B \in \mathbb{R}^{d\times k}$

Initialize the valid-difference list $\mathcal{L}_\Delta \leftarrow [\,]$
for all $x_i \in \mathcal{D}_{\text{basis}}$ **do**
  $\mathcal{C}_i \leftarrow \mathcal{G}(f, x_i)$
  if x_i is a captioning input **then**
    Score object mentions in $\mathcal{C}_i$ using the annotations
    Select a grounded caption y_i^+ and the highest-hallucination caption y_i^-
  else
    Determine the queried object's presence from the annotations
    Select $y_i^+, y_i^- \in \mathcal{C}_i$ giving correct and incorrect existence judgments, respectively
  end if
  if no valid pair exists **then**
    continue
  end if
  for $s \in \{+,-\}$ **do**
    for $t = 1, \ldots, T_i^s$ **do**
      $h_{i,t}^s \leftarrow f_{\le\ell}(x_i, y_{i,\le t}^s)$
    end for
  end for
  if either answer-state sequence is empty or invalid **then**
    continue
  end if
  for $s \in \{+,-\}$ **do**
    $\bar{h}_i^s \leftarrow \dfrac{1}{T_i^s}\displaystyle\sum_{t=1}^{T_i^s} h_{i,t}^s$
  end for
  $\Delta_i \leftarrow \bar{h}_i^- - \bar{h}_i^+$
  Append Δ_i to $\mathcal{L}_\Delta$
end for
$n \leftarrow |\mathcal{L}_\Delta|$
if $n = 0$ or $k < 1$ or $k > \min(n, d)$ **then**
  stop with an error
end if
Index the elements of $\mathcal{L}_\Delta$ as $\Delta_1, \ldots, \Delta_n$ and form $D \leftarrow [\Delta_1, \ldots, \Delta_n]^\top$
$U\Sigma V^\top \leftarrow \mathrm{SVD}(D)$
return $B \leftarrow V_{:,1:k}$

of these differences, including both their mean and cross-example variation. Empty answers and failed hidden-state evaluations are skipped. The leading k right singular vectors form the frozen orthonormal basis B.

D.2 Navigator Training

Algorithm 2 evaluates each candidate under teacher forcing, using the observed response prefix $y_{<t}$ while retaining the KV cache produced by preceding SCAN interventions within that candidate. For compactness, let $\mathsf{Edit}_{\theta,\alpha}(h, B)$ denote the SCAN update derived in Eq. (28).

$$
\begin{aligned}
c = B^\top h, \qquad & z = h - Bc, \\
(a, \rho) = g_\theta(z), \qquad & \lambda = \mathrm{softplus}(\rho), \\
\gamma = \frac{\alpha\lambda}{1 + \alpha\lambda}&, \\
\mathsf{Edit}_{\theta,\alpha}(h, B) &= h - B[\gamma \odot (c - a)].
\end{aligned}
$$

Algorithm 2 Navigator Training

Input Frozen f and B, layer ℓ, preference sets $\mathcal{D}_{\text{cap}}$ and $\mathcal{D}_{\text{exist}}$, training strength α_{train}, steps S
Output Trained Navigator parameters θ

Initialize θ and freeze all parameters of f and B
Cache reference scores for every candidate under $\pi_0 = f$
for $s = 1, \ldots, S$ **do**
  Sample task-balanced batches $\mathcal{B}_{\text{cap}}$ and $\mathcal{B}_{\text{exist}}$
  for all candidate (x, y) in the sampled preference pairs **do**
    Initialize a fresh KV cache for this candidate
    for $t = 1, \ldots, |y|$ **do**
      $h_t^{(\ell)} \leftarrow f_{\le \ell}(x, y_{<t})$
      $\tilde{h}_t^{(\ell)} \leftarrow \mathsf{Edit}_{\theta, \alpha_{\text{train}}}(h_t^{(\ell)}, B)$
      Continue through $f_{>\ell}$ to obtain $\log \pi_\theta(y_t \mid x, y_{<t})$
      Retain the resulting KV cache for the next prediction
    end for
    Compute the object-weighted, length-normalized sequence score $S_\theta(y \mid x)$
  end for
  Form $\widehat{\mathcal{L}}_{\text{cap}}$, $\widehat{\mathcal{L}}_{\text{exist}}$, and $\widehat{\mathcal{L}}_{\text{reg}}$ from the sampled pairs
  $\widehat{\mathcal{L}}_{\text{nav}} \leftarrow \widehat{\mathcal{L}}_{\text{cap}} + \widehat{\mathcal{L}}_{\text{exist}} + \widehat{\mathcal{L}}_{\text{reg}}$
  $g \leftarrow \nabla_\theta \widehat{\mathcal{L}}_{\text{nav}}$
  $\theta \leftarrow \mathrm{AdamW}(\theta, g)$
end for
return θ

The division in the expression for γ is elementwise. The map is applied to the current final-position hidden state used to predict the next answer token. For the first answer token, this is the final prompt position; subsequent predictions use the position of the preceding answer token. Other image and prompt positions are left unchanged.

Navigator training is built from the full 5,334-pair preference corpus, with the fixed 4,267-pair optimization split used for gradient updates and the remaining splits reserved for diagnostics and held-out evaluation. Each optimization step samples four captioning pairs and four object-existence pairs independently with replacement. The captioning, object-existence, and displacement terms are summed as in the main-paper objective and optimized directly with AdamW.

Reference scores are computed once with the frozen, unmodified backbone. Sequence scores cover answer tokens and use the object-weighted, length-normalized scoring rule from the main paper. Object-aligned positions receive the object multiplier, while all remaining answer positions receive unit weight. Each candidate is evaluated with its own KV cache, so historical interventions affect subsequent predictions within that candidate. Gradients pass through the frozen backbone computation graph into the Navigator while the backbone and basis remain fixed.

D.3 Autoregressive Inference

Algorithm 3 gives the autoregressive decoding procedure. SCAN edits $h_t^{(\ell)}$ between $f_{\le \ell}$ and $f_{>\ell}$ at the current prediction position. Inference retains the KV cache across prediction steps, as in teacher-forced training, and extends the prefix with the generated token at each step.

The implementation realizes current-position editing with a post-block forward hook and standard KV caching. During prompt prefill, the edit is applied to the final prompt position to predict the first answer token. Because the hook acts after block ℓ, that block retains its current-position K/V tensors, while downstream blocks construct their cache entries from the edited state. Later decoding steps apply the same edit to the current final position and retain the accumulated cache.

Algorithm 3 Autoregressive Decoding with SCAN

Input Image–prompt input x, frozen L-layer LVLM f, intervention layer ℓ, basis B, Navigator g_θ, strength α, decoding rule $\mathcal{A}$, maximum length T
Output Generated token sequence y

Initialize the generated prefix $y_{<1} \leftarrow \varnothing$
Initialize a fresh KV cache
for $t = 1, \ldots, T$ **do**
  $h_t^{(\ell)} \leftarrow f_{\le \ell}(x, y_{<t})$
  $\tilde{h}_t^{(\ell)} \leftarrow \mathsf{Edit}_{\theta,\alpha}(h_t^{(\ell)}, B)$
  $o_t \leftarrow f_{>\ell}(\tilde{h}_t^{(\ell)}, x, y_{<t})$
  Retain the resulting KV cache for the next prediction
  $y_t \leftarrow \mathcal{A}(o_t)$
  $y_{<t+1} \leftarrow (y_{<t}, y_t)$
  if $y_t = \mathrm{EOS}$ **then**
    break
  end if
end for
return $y_{<t+1}$

Table 8: Navigator architecture and initialization for LLaVA-1.5 7B.

Component	Specification
Architecture	
Dimensions	$d = 4096$, $k = 128$, trunk width $m = 1024$
Input	Residual $z_t \in \mathbb{R}^d$
Normalization	LayerNorm over d features
Shared trunk	Linear(d, m), SiLU, dropout 0.05
Anchor head	Linear(m, k) producing a_t
Direction-weight head	Linear(m, k) producing ρ_t; $\lambda_t = \mathrm{softplus}(\rho_t)$
Total output	$2k$ scalars from the two heads
Initialization	
LayerNorm	Scale 1; bias 0
Shared trunk	Xavier-uniform weights; bias 0
Anchor head	Weights 0; bias 0
Direction-weight head	Weights 0; bias -1.35

D.4 Navigator Architecture and Complexity

Table 8 specifies the Navigator architecture and initialization for the primary LLaVA-1.5 7B configuration. The anchor and direction-weight heads share one trunk and have independent output parameters. Other backbones use the same architecture with their respective hidden dimension d.

Let the trunk width be m (implemented as $m = 1024$). Including biases and the affine LayerNorm, the trainable parameter count is

$$P_{\mathrm{nav}} = 2d + (dm + m) + 2(mk + k). \tag{37}$$

For $d = 4096$ and $k = 128$, the three contributions are $8{,}192$, $4{,}195{,}328$, and $262{,}400$, giving exactly $4{,}465{,}920$ trainable parameters.

For one edited position, computing $B^\top h_t$, reconstructing Bc_t, and mapping the coordinate update back through B each cost $O(dk)$. LayerNorm costs $O(d)$, the shared trunk costs $O(dm)$, and the two heads cost $O(2mk)$. The added arithmetic is therefore $O(3dk + dm + 2mk + d)$ per decoding step, with storage $O(dk + P_{\mathrm{nav}})$ for the frozen basis and Navigator. SCAN adds one lightweight Navigator and several matrix–vector products, but no second LVLM branch or additional backbone forward pass. Measured throughput is reported with the efficiency experiments.

E Residual Conditioning and Generation Stability

SCAN predicts the anchor and direction-wise weights from the orthogonal residual, while using the current subspace coordinates to compute the correction. This section characterizes the local constraints imposed by this input choice and compares residual and full-state conditioning under the same training protocol. The empirical comparison focuses on repetition during autoregressive generation and interprets CHAIR jointly with object coverage.

E.1 Local Properties of Residual Conditioning

At the current decoding position and intervention layer, the decomposition in Eq. (1) gives

$$h_t = Bc_t + z_t, \qquad c_t = B^\top h_t, \qquad z_t = (I_d - BB^\top)h_t. \tag{38}$$

Here, Bc_t is the component that SCAN edits, and z_t is the orthogonal residual of the representation induced by the image, prompt, and response prefix. Since $B^\top B = I_k$, any perturbation confined to the editable subspace satisfies

$$(I_d - BB^\top)(h_t + B\delta) = (I_d - BB^\top)h_t = z_t, \qquad \delta \in \mathbb{R}^k. \tag{39}$$

Consequently, changing c_t while holding z_t fixed leaves the residual Navigator's input unchanged. At inference, its predicted anchor and weights therefore remain unchanged, while c_t determines the displacement toward the anchor.

A full-state Navigator allows the anchor and weights to depend on both components:

$$\begin{aligned} (a_t, \rho_t) &= g_\theta(Bc_t + z_t), \\ \lambda_t &= \mathrm{softplus}(\rho_t), \\ \gamma_t &= \frac{\alpha\lambda_t}{1 + \alpha\lambda_t}, \\ \widetilde{c}_t &= c_t - \gamma_t \odot (c_t - a_t). \end{aligned} \tag{40}$$

Both input choices preserve the current orthogonal residual because the correction lies in $\mathrm{span}(B)$. Residual conditioning additionally makes the predicted anchor and weights invariant to changes in the current editable coordinates. Both variants predict the Navigator outputs once and apply one correction at the selected layer.

This restriction also gives a local nonexpansiveness property. For fixed Navigator parameters, residual z, and global strength α, define $\Gamma(z) = \mathrm{diag}(\gamma(z))$. The inference-time coordinate update is

$$\Phi_z(c) = (I_k - \Gamma(z))c + \Gamma(z)a(z). \tag{41}$$

Since $0 \le \gamma_j(z) < 1$, any two coordinate vectors sharing the same residual satisfy

$$\begin{aligned} \|\Phi_z(c_1) - \Phi_z(c_2)\|_2 &= \|(I_k - \Gamma(z))(c_1 - c_2)\|_2 \\ &\le \|c_1 - c_2\|_2. \end{aligned} \tag{42}$$

Thus, at a fixed residual, the update does not amplify differences in the editable coordinates. For a full-state Navigator, the anchor and gains may also change with c, so the gain bounds alone do not imply the same property.

These local properties do not require the residual to contain all information relevant to target selection. As established in Appendix B.5, richer conditioning information cannot increase the optimal squared-error approximation risk under a fixed distribution and unrestricted prediction rules. SCAN uses residual conditioning as a structural restriction on the learned predictor. Its practical value under preference training and autoregressive decoding is assessed empirically below.

E.2 Intervention Effects Across Decoding Steps

An intervention can influence later generation through downstream computation and the generated prefix. Under the single-layer post-block implementation in Appendix D, an edit changes the remaining layer computations, the cache entries formed by downstream blocks, and the next-token

distribution. At a fixed token prefix, these downstream cache entries affect later downstream computations but do not feed back into the pre-edit state at layer ℓ. During free-running decoding, a change in the selected token changes the prefix and can therefore change subsequent Navigator inputs, including their residuals.

Both teacher-forced training and free-running decoding retain the cache produced by preceding interventions within a response. Training supplies the observed response tokens, whereas free-running decoding extends the prefix with tokens selected by the intervened model. The Navigator is therefore evaluated on generated prefixes at inference, where both its residual input and the editable coordinates may differ from those encountered during training.

The invariance and nonexpansiveness above apply locally at a fixed residual. Across decoding steps, the residual, anchor, and gains can all change. Likewise, the coordinate-wise interpolation property in Appendix C constrains each correction relative to its current anchor. These properties motivate evaluating generation behavior, but do not establish sequence-level stability or identify the cause of a particular repetition pattern.

E.3 Controlled Comparison of Navigator Inputs

We compare residual-only and full-state Navigators on the same 500 MS COCO images with LLaVA-1.5 7B. Both variants use the same rank-128 basis, intervention layer 31, 5,334-pair preference corpus, fixed 4,267-pair optimization split, random seed, and 500 training steps. Generation uses greedy decoding with a 64-token limit. The conditioning input is the only architectural change.

At $\alpha = 0.10$, the comparison holds the global strength fixed while retaining the training and evaluation settings. The learned anchors and direction-wise weights may differ between the two variants, so equal α does not imply matched gains or hidden-state displacements. Additional full-state results at $\alpha = 0.02$ and 0.05 characterize behavior at lower global strengths; these operating points are not matched to the residual-only setting for object recall or displacement magnitude.

We mark a generation as a repetition loop when the same normalized word appears at least three times consecutively. Every detected loop in this study repeats the word "another." Table 9 reports loop counts and rates together with object mentions, recall, mean output length, and CHAIR. Recall and CHAIR are reported as percentages. Gray CHAIR scores marked by ‡ identify settings with detected repetition. Repetition under this criterion is the primary stability measure; CHAIR is interpreted jointly with loop rates and object coverage.

Table 9: Generation behavior under different Navigator inputs.

Input	α	Loops (rate)↓	Object mentions	Recall↑	Mean length	$C_s \downarrow$	$C_i \downarrow$
Residual z_t	0.10	0/500 (0.0%)	1,693	63.96	46.9	10.00	4.02
Full h_t	0.02	25/500 (5.0%)	2,120	67.63	49.3	$10.80^{\ddagger}$	$3.82^{\ddagger}$
Full h_t	0.05	355/500 (71.0%)	1,517	60.28	53.1	$6.00^{\ddagger}$	$2.44^{\ddagger}$
Full h_t	0.10	468/500 (93.6%)	942	51.76	55.9	$2.00^{\ddagger}$	$1.27^{\ddagger}$

E.4 Empirical Analysis of Generation Stability

At the common global strength $\alpha = 0.10$, no repetition loops are detected for the residual-only Navigator, whereas the full-state Navigator loops on 468 of 500 images (93.6%). The full-state results at $\alpha = 0.05$ and 0.02 have lower loop rates of 71.0% and 5.0%, respectively. These measurements support residual conditioning as the more stable choice under the reported training protocol and operating points.

The lowest full-state strength presents a trade-off between coverage and repetition. Relative to the residual-only setting at $\alpha = 0.10$, full-state conditioning at $\alpha = 0.02$ has higher recall and slightly lower C_i, alongside higher C_s and repetition on 25 images. This result distinguishes the limited repetition observed at the lower strength from the widespread degeneration at larger strengths.

Across the full-state operating points from $\alpha = 0.02$ to $\alpha = 0.10$, object mentions decrease from 2,120 to 942 and recall falls from 67.625% to 51.755%, while mean output length increases from 49.254 to 55.928. Together with the rising loop rate, these measurements show that longer outputs can contain fewer object descriptions. They also show that lower CHAIR can coexist with severe repetition and reduced object coverage. The lower CHAIR scores therefore cannot be interpreted as improved grounding in isolation.

Table 10 illustrates the observed repetition on matched images. Repetition can begin after a generic opening or after an initially descriptive passage. The examples document the form of the degeneration and complement the aggregate measurements; they do not isolate its underlying mechanism.

The structural analysis and empirical comparison support residual conditioning for the evaluated configuration. The analysis identifies the dependence excluded by this design: the predicted anchor and weights do not vary directly with the current editable coordinates. The experiment establishes the observed repetition advantage under the reported protocol.

Table 10: Examples of repetition in matched generations. Bracketed text summarizes repeated occurrences of "another," and ellipses indicate omitted text. The examples illustrate generation behavior and do not provide independent grounding annotations.

Image	Residual-only z_t	Full-state h_t
272694	The image showcases a dining table with a variety of food and beverages on it. A Starbucks coffee cup is positioned towards the left side of the table, while a muffin and a banana are situated closer to the right side. Additionally, the scene includes …	The image displays a ["another" repeated 60 times].
146126	The image showcases a close-up of a meal on a dining table. The meal consists of a bowl of noodles and a lighter meal of a salad. The noodles are accompanied by a beverage in a cup, while the salad is placed in …	The image displays ["another" repeated 59 times].
350289	The image captures a beautiful beach scene with two surfers walking along the shore. Both surfers are carrying their surfboards as they make their way towards the ocean. The waves …	The image captures a beautiful beach scene with two men prominently featured. One man is holding a surfboard ["another" repeated 40 times].

F Data Construction and Preprocessing

F.1 Data Sources and Splits

The basis-construction and Navigator preference pools are built from MS COCO images and object annotations. Candidate responses are generated by frozen LVLMs and organized into preferred and rejected pairs using annotation-based criteria. Each retained pair corresponds to a distinct image within its pool, so the image and pair counts are equal.

The basis pool contains 3,072 pairs and is used only to estimate paired representation differences. The Navigator corpus contains 5,334 pairs, partitioned into 4,267 training pairs, 533 development pairs, and 534 held-out test pairs. Only the training split is used for gradient updates; the develop-

Table 11: Composition of the basis and Navigator data pools.

Data partition	Pairs	Captioning	Object existence
Basis pool	3,072	2,304	768
Navigator pool	5,334	2,657	2,677
Training	4,267	2,126	2,141
Development	533	262	271
Test	534	269	265

ment and test splits support training diagnostics and held-out preference evaluation, respectively. Table 11 summarizes the task composition.

The basis and Navigator pools are checked for overlap using pair identifiers, normalized response content, resolved image paths, and COCO image identifiers. No cross-pool overlap is found under these checks, leaving the two pools disjoint at both the image and response levels.

The mechanism analysis uses a separately constructed MS COCO dataset containing 1,000 held-out grounded and hallucinated response pairs, distinct from the Navigator test split. The 500 MS COCO images used for CHAIR evaluation are explicitly excluded when constructing the basis pool, all three splits of the Navigator corpus, and the mechanism-analysis dataset. The same CHAIR image set is used for both the 64- and 512-token evaluation protocols.

F.2 Pair Construction and Token Weighting

For captioning, frozen LVLMs generate candidate responses under multiple decoding configurations. We evaluate all candidates using MS COCO annotations and the CHAIR object inventory. Each preferred caption contains no unsupported object mention under these criteria, while its rejected counterpart contains at least one. Pair selection prioritizes similar annotated-object recall and descriptive content to reduce differences unrelated to grounding. Images without a valid grounded–hallucinated pair are excluded. Grounding labels are defined with respect to the adopted object inventory and annotations.

For object existence, examples use the question template `Is there a/an {object} in the image?` For the same image and queried object, the preferred answer gives the correct existence judgment according to the MS COCO annotations, while the rejected answer expresses the opposite judgment. Both answers are complete natural-language responses. Preference labels depend on whether the answer correctly affirms or denies the object's presence.

Both tasks use the same object-token weighting procedure. Object mentions are normalized using the CHAIR synonym inventory and aligned to answer tokens, with longer phrases matched before shorter ones. All subword tokens covered by a matched mention receive weight $w_{\mathrm{obj}} = 3$; the remaining answer tokens receive weight 1. This rule applies to both preferred and rejected responses, and the intervened model π_θ and reference π_0 use identical weights for each response. Sequence scores cover answer tokens only and normalize the weighted sum of token log probabilities by the sum of their weights. When no object mention is detected, the score reduces to the standard length-normalized answer log likelihood.

G Hyperparameters and Sensitivity Analysis

G.1 Default Configuration

The main LLaVA-1.5 7B configuration uses a rank-128 subspace at layer 31. We optimize only the Navigator, using the 4,267-pair training split described in Appendix F, while keeping the backbone and basis frozen. Table 12 summarizes the default optimization settings. The default inference strength is $\alpha = 0.10$.

At step 125, the learning rate is reduced by a factor of four, and the displacement coefficient η_h changes from 0 to 0.04. The objective combines the captioning, object-existence, and displacement

Table 12: Default Navigator training settings.

Setting	Value
Optimizer	AdamW
Optimization steps	500
Pairs per step	4 captioning + 4 existence
Learning rate	$3.0 \times 10^{-4} \rightarrow 7.5 \times 10^{-5}$ (step 125)
Weight decay	0
Gradient clipping	1.0
Caption preference β_{cap}	0.30
Existence preference β_{exist}	0.10
Displacement coefficient η_h	$0 \rightarrow 0.04$ (step 125)
Object-token weight w_{obj}	3
Random seed	0

Table 13: Sensitivity to training and representation settings.

Factor	Setting	$\mathrm{C}_s \downarrow$	$\mathrm{C}_i \downarrow$
Default	**Reference configuration**	**10.00**	**4.02**
Subspace rank	$k = 64$	13.20	5.39
	$k = 256$	11.20	4.10
Intervention layer	Layer 23	12.80	4.53
	Layer 27	11.60	4.21
Training steps	250	15.20	6.21
	750	10.80	4.07
Initial learning rate	1.5×10^{-4}	16.20	6.96
	6.0×10^{-4}	12.60	3.99

terms defined in the main paper. The default configuration uses the checkpoint at step 500. Training takes approximately 50 minutes on one NVIDIA RTX 4090.

G.2 Sensitivity Analysis

All sensitivity studies use LLaVA-1.5 7B and the fixed 500-image MS COCO set used for the 64-token CHAIR evaluation. The preference data, random seed, and greedy decoding protocol are held fixed. CHAIR and object recall are reported as percentages.

Training and representation settings. We evaluate the subspace rank, intervention layer, training duration, and initial learning rate. Each variant changes one training or representation setting, with its inference strength selected to approximate the default object recall. Table 13 reports the resulting CHAIR scores and highlights the default configuration in bold.

For the learning-rate variants, the initial learning rate is varied while the fourfold decay at step 125 is retained. For the training-duration variants, both the learning-rate decay and the activation of displacement regularization remain at step 125; only the stopping step changes.

Among the evaluated operating points, the default configuration achieves the lowest C_s. Both alternative ranks and earlier intervention layers yield higher values of both CHAIR metrics. Reducing training to 250 steps worsens performance, while extending it to 750 steps provides no improvement over the default. Halving the initial learning rate produces the largest increase in both metrics. Doubling it yields a slightly lower C_i (3.99 versus 4.02), alongside a higher C_s (12.60 versus 10.00).

Inference-time navigation strength. We compare SCAN with a zero-target baseline that uniformly shrinks the editable subspace coordinates. For a scalar strength $s \in [0, 1]$, the baseline ap-

Table 14: Hallucination–recall trade-offs across intervention strengths.

Method	Strength	$C_s \downarrow$	$C_i \downarrow$	Recall↑
Vanilla	0.000	20.40	7.08	71.12
SCAN	0.010	19.20	6.59	70.50
	0.020	18.80	6.88	70.04
	0.040	15.20	5.55	68.25
	0.060	13.20	4.95	65.63
	0.080	12.80	4.76	65.40
	0.100	**10.00**	**4.02**	**63.96**
	0.150	8.80	4.11	60.66
Zero target	0.375	18.80	6.63	69.13
	0.450	17.80	6.58	68.18
	0.525	17.60	6.77	65.78
	0.600	15.20	6.26	64.80
	0.700	12.40	5.72	61.80

plies

$$\tilde{h}_t^0 = h_t - sBB^\top h_t = z_t + B[(1-s)c_t]. \tag{43}$$

The coefficient s directly specifies the fraction of the subspace component removed at each edited position. The same scalar is applied to all subspace directions and decoding steps.

Table 14 reports the measured operating points for both methods. The strength column denotes the global navigation parameter α for SCAN and the direct shrinkage coefficient s for the zero-target baseline. All points use the same images and 64-token CHAIR protocol. Reporting recall alongside CHAIR allows hallucination reduction to be assessed together with changes in object coverage.

These measurements underlie the trade-off curves in Figure 1(b) of the main paper. Bold values mark the default SCAN operating point.

Across the measured strengths, both methods show decreasing sentence-level CHAIR and object recall, while instance-level CHAIR varies non-monotonically. At nearly matched recall, SCAN with $\alpha = 0.06$ achieves $C_s = 13.20$ and $C_i = 4.95$ at 65.63% recall. The zero-target baseline with $s = 0.525$ obtains $C_s = 17.60$ and $C_i = 6.77$ at 65.78% recall. The recall difference is 0.15 percentage points, while SCAN has lower C_s and C_i by 4.40 and 1.82 percentage points, respectively. This comparison supports a more favorable hallucination–recall trade-off at these operating points.

H Evaluation Protocol

H.1 Prompts and Model Formatting

The 64- and 512-token CHAIR evaluations use the prompt `Please describe this image in detail.` The qualitative study uses the same instruction. AMBER uses `Describe this image in detail.` For POPE, we append `Please answer this question with one word.` to each official benchmark question.

These task instructions are shared across backbones. Each model applies its native conversation template and multimodal tokenization, with image placeholders and role markers following the corresponding model interface.

H.2 Decoding and Metric Computation

CHAIR uses a fixed set of 500 MS COCO images, sampled once with a fixed random seed and shared by the 64- and 512-token evaluations. Its separation from the data used for basis construction, Navigator development, and mechanism analysis is described in Appendix F. Captions are generated greedily and stop at EOS or the maximum generation length of 64 or 512 new tokens, depending on the evaluation setting. Sentence-level CHAIR (C_s), instance-level CHAIR (C_i), and

annotated-object recall are computed using the official CHAIR object inventory, synonym mapping, and singularization rules.

POPE is evaluated with batch size one. Each complete generated response is lowercased and parsed for a yes/no decision before comparison with the ground-truth label. An unambiguous standalone occurrence of "yes" is mapped to a positive prediction, and an unambiguous standalone occurrence of "no" is mapped to a negative prediction. Responses containing neither decision word, or both words without an unambiguous decision, are counted as incorrect. Accuracy and F1 are reported separately for the Random, Popular, and Adversarial splits.

AMBER uses the official generative evaluation set. Newline characters in generated responses are replaced with spaces before the official evaluator computes CHAIR, Cover, Hal, and Cog. Apart from this newline normalization for AMBER, generated captions are passed directly to the respective CHAIR and AMBER evaluation tools.

Efficiency is measured with LLaVA-1.5 7B at batch size one on the same 500 CHAIR images. Each sample generates exactly 64 new tokens with EOS stopping disabled. After 20 warm-up generations, each method is measured over three complete runs. Timing includes visual encoding and autoregressive decoding, with CUDA synchronization immediately before and after generation. We report throughput in generated tokens per second and relative speed as a percentage of Vanilla throughput.