DIAGNOSE BEFORE YOU DESIGN: DEPTH–SCALE DEPENDENCIES IN FOUNDATION-VIT DETECTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Detectors built on plain foundation Vision Transformers (ViTs) must construct multiscale spatial memories from single-resolution backbone features, yet existing interfaces typically predefine how backbone depth is mapped to detector scale without directly measuring the resulting functional dependencies. Rather than repeatedly retraining alternative routing structures, we recast interface analysis as an *interventional diagnosis* problem: controlled inference-time interventions on a reference detector quantify the functional dependence of existing depth-to-scale paths and spatial memories. Across DINOv2, DINOv3, and distinct detector interfaces, we consistently observe strong and image-specific depth–scale dependencies, while their dominant spatial realization changes with backbone representation and detector topology. Moreover, weak marginal dependence in a fixed model does not imply retraining-time structural redundancy. Controlled factorial experiments reveal that the effect of depth treatment is modulated by spatial memory topology, establishing the two as coupled design variables and motivating a diagnosis-guided redesign workflow. Guided by these measurements, we formulate and validate distinct interface redesign hypotheses across DINOv3- and DINOv2-based detectors that preserve aggregate Full-COCO accuracy while reducing model capacity, demonstrating the practical value of diagnosis-guided interface design.

1 INTRODUCTION

Self-supervised foundation Vision Transformers such as DINOv2 (Oquab et al., 2024) and DINOv3 (Siméoni et al., 2026) have emerged as powerful general-purpose visual backbones, offering strong semantic representations together with rich spatial information. Unlike convolutional networks, however, plain ViTs do not natively provide a multiresolution feature hierarchy. To adapt them to dense prediction tasks such as object detection, modern detectors—including LW-DETR (Chen et al., 2024), RF-DETR (Robinson et al., 2026), and DEIMv2 (Huang et al., 2025a)—construct multiscale representations by resampling and fusing intermediate ViT features. This interface must make two architectural choices: which backbone depths to expose and how those features should be assigned to detector spatial memories. Existing systems typically make these choices through predefined layer indices, fixed scale mappings, and heuristic fusion structures, yet it remains unclear whether these design decisions reflect the functional dependencies that the trained detector ultimately develops.

A natural way to study such interfaces is to modify the routing structure, retrain each candidate detector from scratch, and select among the resulting accuracy–efficiency trade-offs. As shown in Figure 1(a), this conventional train-and-compare strategy has three limitations. First, every candidate incurs a separate training cost. Second, practical compute budgets restrict exploration to only a small subset of possible interface choices. Third, final detection metrics can rank candidate architectures but provide no direct path-level evidence about which depth-to-scale paths or spatial memories the trained detector actually relies on. We therefore recast interface analysis as a functional diagnosis problem on a reference detector. Keeping the model structure and parameters fixed, we intervene on existing depth-to-scale paths and spatial memories and measure the resulting changes in detection performance. Each intervention requires only forward evaluation, and the basic diagnostic cost scales approximately linearly with the size of the existing interface rather than with the combinatorial space of candidate architectures.

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

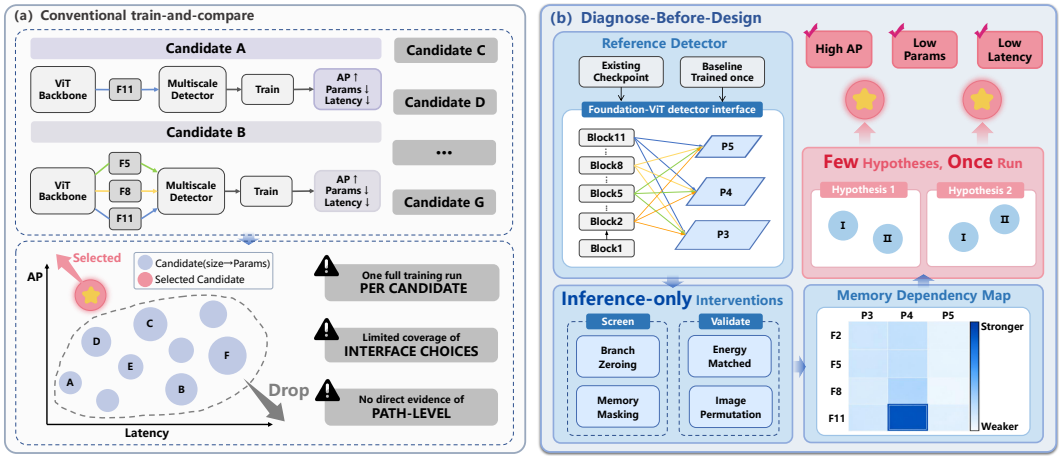

Figure 1: **Diagnose-Before-Design versus conventional train-and-compare.** Conventional interface search repeatedly retrains candidates and relies on endpoint metrics, making exploration costly and offering little path-level evidence. Diagnose-Before-Design instead uses inference-only interventions to expose functional dependencies and focus retraining on a few evidence-driven redesigns.

Applying this diagnosis across different backbones and detector interfaces reveals a consistent but non-universal structure. Foundation-ViT detectors develop highly asymmetric and image-specific depth–scale dependencies, but the dominant spatial realization changes with the system. In a representative DINOv3-S detector equipped with an RF-DETR-style multiscale projector (MSP), the deepest backbone representation is used primarily through the $F_{11} \rightarrow P_4$ path. Keeping the MSP topology fixed but replacing the backbone with native DINOv2-S/14 shifts the dominant destination to P_5. In DEIMv2, the deepest semantic representation remains critical, but it is propagated through a qualitatively different, P_5-centered internal top-down carrier. These comparisons suggest a broader principle: structured deep-semantic dependence is recurrent, whereas its spatial realization is system-dependent.

Motivated by these observations, we introduce *Diagnose-Before-Design*: starting from an existing checkpoint or a baseline trained once, inference-only interventions expose the functional dependency structure of the interface, which is then used to formulate a small set of redesign hypotheses for controlled retraining (Figure 1(b)). The workflow therefore moves functional evidence before expensive candidate-wise train-and-compare rather than replacing architecture validation.

Our contributions are threefold:

1. **A lightweight interventional diagnosis protocol for ViT detector interfaces.** We formulate depth-to-scale interface analysis as functional diagnosis of a trained detector and combine source-branch zeroing, spatial-memory masking, energy-matched controls, and cross-image permutation to measure path- and memory-level dependencies without retraining every candidate routing structure.
2. **System-dependent depth–scale dependence and depth–topology coupling.** Across DINOv3, DINOv2, and DEIMv2, we consistently observe strong image-specific deep-semantic dependence, while its dominant spatial realization changes with backbone representation and detector topology. Controlled factorial experiments further show that spatial-memory topology modulates the effect of depth treatment.
3. **Diagnosis-guided interface redesign at the appropriate structural granularity.** The resulting dependency maps motivate system-specific redesign strategies: coupled topology–depth restructuring for DINOv3-based detection and targeted source-branch pruning for DINOv2-based detection. Full-COCO experiments confirm that both directions preserve aggregate detection accuracy while reducing model capacity.

2 Related Work

Foundation Vision Transformers for Dense Prediction. Plain Vision Transformers have become strong general-purpose visual backbones, spanning the original ViT architecture (Dosovitskiy et al., 2021) and self-supervised pretraining approaches such as MAE (He et al., 2022), DINO (Caron et al., 2021), DINOv2 (Oquab et al., 2024), and DINOv3 (Siméoni et al., 2026). Hierarchical architectures obtain multiscale representations explicitly: convolutional networks expose progressively downsampled stages that can be organized into feature pyramids (Lin et al., 2017), while hierarchical Transformers such as Swin (Liu et al., 2021) introduce progressive spatial reduction within the backbone itself. Plain ViTs, by contrast, maintain a largely uniform token resolution across depth while their internal representations evolve substantially across layers. Prior analyses show that ViT representations differ systematically from convolutional hierarchies (Raghu et al., 2021) and that deep self-supervised ViT features retain strong local correspondence and dense visual structure (Caron et al., 2021; Amir et al., 2021; Park et al., 2023). iBOT (Zhou et al., 2022) provides complementary evidence of emergent local semantics and strong transfer to dense prediction. These properties make foundation ViTs attractive for dense prediction, but their non-hierarchical structure leaves open a practical interface question: which intermediate depths should feed which detector scales?

Feature Pyramids and Adapters for Plain ViTs. A broad line of work adapts plain ViTs to multiscale dense prediction by constructing spatial hierarchies externally. DPT (Ranftl et al., 2021) reassembles representations from multiple transformer stages into multiresolution feature maps for dense prediction. ViTDet (Li et al., 2022) instead builds a simple feature pyramid from a single-scale final ViT representation. Not all dense-prediction interfaces construct an explicit feature pyramid: Segmenter predicts semantic masks directly from ViT patch embeddings without an explicit pyramid (Strudel et al., 2021), contrasting with pyramid-style adaptation. ViT-Adapter (Chen et al., 2023), LW-DETR (Chen et al., 2024), and RF-DETR (Robinson et al., 2026) likewise adapt intermediate transformer features into detector-scale representations through learned adaptation, projection, or resampling modules. DEIMv2 (Huang et al., 2025a) introduces a Spatial Tuning Adapter (STA) that combines resized DINOv3 features with lightweight convolutional spatial priors before further multiscale processing. These designs are effective, but they largely specify backbone-depth selection, scale assignment, and detector topology as architectural choices. Our work instead studies how those choices relate to the functional dependencies that emerge after training, and whether depth treatment can be separated from spatial-memory topology.

Representation Probing and Interventional Model Analysis. A complementary literature seeks to understand what neural representations encode and how they are used. Network Dissection (Bau et al., 2017) links hidden units to semantic concepts, while representation-similarity methods such as centered kernel alignment (CKA) (Kornblith et al., 2019) compare internal feature structures across networks and depths. Analyses of self-supervised ViTs (Caron et al., 2021; Park et al., 2023) have likewise characterized how semantic and spatial structure evolves across transformer depth. Such probing and similarity analyses primarily characterize what information is present in a representation. Interventional analyses instead manipulate internal activations and measure the resulting output changes; prior work (Bau et al., 2019) has used such interventions to quantify the functional effect of internal representation units. We adopt this interventional perspective at the foundation-ViT detector interface, treating depth-to-scale paths and spatial memories as the units of analysis instead of individual neurons or isolated layers.

Efficient DETRs and Interface Simplification. DETR (Carion et al., 2020) established end-to-end object detection with transformer decoders, while Deformable DETR (Zhu et al., 2021) introduced multiscale deformable attention to improve convergence and dense localization. More recent systems such as RT-DETR (Zhao et al., 2024), D-FINE (Peng et al., 2025), and DEIM (Huang et al., 2025b) further improve real-time detection through more efficient feature encoding, localization, and training strategies. Sparse DETR reduces detector-side computation by selectively updating encoder queries (Roh et al., 2022). In these architectures, computational cost depends not only on the decoder itself but also on the number and size of multiscale spatial memories exposed to it. Most efficiency-oriented work, however, focuses on the detection head, decoder, or training mechanism rather than diagnosing redundancy in the upstream foundation-ViT interface. Our Diagnose-Before-Design approach targets this complementary level: it uses interventional evidence from the trained interface to guide topology simplification and depth-aware restructuring.

3 DIAGNOSING THE VIT–DETECTOR INTERFACE

3.1 INTERVENTIONAL DEPTH–SCALE DIAGNOSIS

Let $\mathcal{F} = \{F_d\}_{d\in\mathcal{D}}$ denote the intermediate features extracted from a plain-ViT backbone at selected depths d, and let $\mathcal{P} = \{P_s\}_{s\in\mathcal{S}}$ denote the multiscale spatial memories consumed by the detector, where $s \in \{3, 4, 5\}$ corresponds to strides 8, 16, and 32, respectively. The interface consists of a set of existing depth-to-scale branches $\mathcal{E} = \{e_{d,s}\}$, followed by feature fusion and memory construction. We focus on a complementary and directly measurable question: which existing paths and spatial memories does a trained detector actually rely on, and how strongly?

For an existing branch $e_{d,s}$, we define its interventional dependence as the performance change induced by a controlled intervention: $D_{\text{edge}}(d, s) = \text{AP}_{\text{base}} - \text{AP}_{\text{int}}(e_{d,s})$. Analogously, we define the dependence of a spatial memory P_s as $D_{\text{mem}}(s) = \text{AP}_{\text{base}} - \text{AP}_{\text{int}}(P_s)$.

Throughout the paper, *interventional dependence* refers to the functional change in detector output caused by manipulating an internal path or memory while keeping model parameters and the evaluation distribution fixed. We measure this dependence using two screening interventions followed by two targeted validation operations.

Source-Branch Zeroing. For each existing branch $e_{d,s}$, we zero its resampled activation while keeping all other paths and parameters fixed. The resulting AP drop measures marginal functional dependence on that path.

Spatial-Memory Masking. We mask all tokens of one memory P_s immediately before decoder cross-attention, leaving upstream encoding and proposal generation unchanged, thereby measuring destination-level dependence.

Energy-Matched Control. For selected dominant paths, we zero the target branch and rescale the remaining inputs to the same destination to match the original aggregate activation norm. Persistence of the AP drop rules out feature-energy loss as the primary explanation.

Fixed-Point-Free Batch Permutation. We permute a selected branch across images within the inference batch, preserving its activation distribution while breaking image correspondence. A larger drop than under zeroing indicates dependence on image-aligned content.

None of these interventions modifies model parameters. If an interface contains $|\mathcal{E}|$ existing depth-to-scale branches and $|\mathcal{S}|$ spatial memories, the basic screening stage requires $O(|\mathcal{E}|+|\mathcal{S}|)$ inference-only evaluations; the energy and permutation controls are applied only to a small set of selected paths. The diagnostic cost therefore scales approximately linearly with the size of the existing interface rather than with the combinatorial space of possible routing subsets.

Controlled analysis setting. To support dense dependency scans, factorial retraining, and repeated control experiments, we freeze a standardized analysis subset, *COCO-Medium*, containing $5{,}000$ training images and $2{,}000$ validation images sampled from COCO 2017 (Lin et al., 2014). Its category frequencies, object-scale distribution, and box-geometry marginals closely track those of Full COCO; the complete representativeness audit is provided in Appendix A. A baseline is trained once in this controlled setting, after which all path interventions reuse the same fixed checkpoint and require only inference. Key dependencies and final architecture endpoints are subsequently confirmed using Full-COCO-trained checkpoints.

Together, the screening and validation interventions produce a functional dependency map of the existing detector interface, which forms the empirical basis for the diagnosis-guided redesigns studied in Section 4.

3.2 ASYMMETRIC AND IMAGE-SPECIFIC DEPENDENCIES IN DINOV3

We first apply the interventional diagnosis to a representative DINOv3-S (Siméoni et al., 2026) + MSP detector, where an RF-DETR-style (Robinson et al., 2026) multiscale projector maps intermediate ViT features into the detector's spatial memories. The resulting dependency map is strikingly asymmetric: rather than distributing functional importance evenly across the predefined depth-to-scale routes, the trained interface concentrates its dependence on a small number of paths.

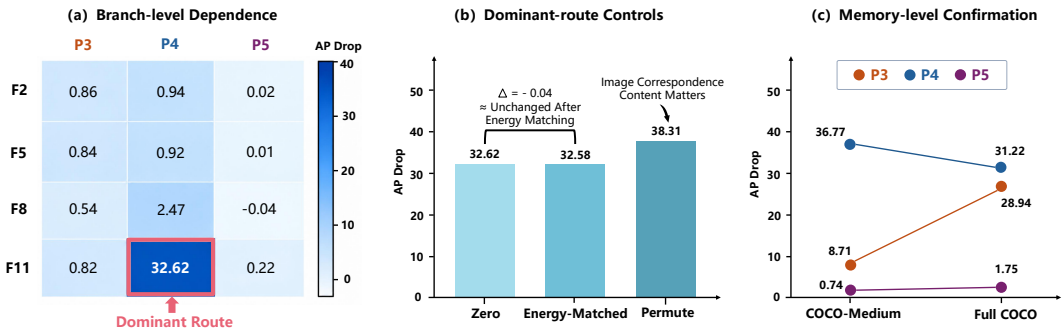

Figure 2: **Interventional diagnosis of DINOv3-S + MSP. (a)** Branch zeroing reveals a highly asymmetric dependency map dominated by $F_{11} \to P_4$. **(b)** Energy matching leaves this effect unchanged, whereas cross-image permutation causes a larger degradation. **(c)** Memory masking confirms strong P_4 dependence and persistently weak P_5 dependence, with P_3 becoming more important after Full-COCO training.

Highly asymmetric branch dependence. Figure 2(a) reports the branch-level dependency map on COCO-Medium. Zeroing the $F_{11} \to P_4$ branch causes a 32.62 AP drop, over 13 times the next-strongest single-branch effect, $F_8 \to P_4$ at 2.47 AP. Most remaining branch interventions change AP by less than 1 point, and every individual branch feeding P_5 has only a minor marginal effect. The trained interface therefore does not use its predefined routes uniformly; instead, deep semantic information is concentrated along a highly specific depth-to-scale path.

Image-specific semantics, not activation magnitude. Figure 2(b) tests whether the dominant $F_{11} \to P_4$ effect can be explained by activation magnitude alone. After removing the branch and rescaling the remaining P_4 inputs to match the original aggregate activation norm, the AP drop remains 32.58, essentially unchanged from the original 32.62. The dominant effect is therefore not explained by a loss of feature energy. We then preserve the branch activation values but permute them across images, thereby breaking correspondence with the current input. This intervention increases the AP drop to 38.31. A mismatched deep feature is thus more disruptive than removing the feature altogether, showing that the detector relies on image-aligned semantic and geometric content rather than on generic deep-feature statistics.

Memory-level confirmation. Figure 2(c) shows that masking P_4 on COCO-Medium causes a 36.77 AP drop, compared with 8.71 for P_3 and only 0.74 for P_5, consistent with the dominant $F_{11} \to P_4$ branch dependence. The same structure persists after full-scale training: on the separately Full-COCO-trained checkpoint, zeroing and cross-image permutation of $F_{11} \to P_4$ cause 25.00 and 47.46 AP drops, while masking P_4, P_3, and P_5 reduces AP by 31.22, 28.94, and 1.75 points, respectively. Thus, P_3 becomes substantially more important at full scale, but the concentrated deep-semantic dependence and weak P_5 endpoint dependence persist.

3.3 System-Dependent Spatial Realization across Backbones and Detector Topologies

Panels (a–b) keep the MSP topology fixed while changing the backbone from DINOv2-S/14 (Oquab et al., 2024) to DINOv3-S (Siméoni et al., 2026), isolating the effect of backbone representation. Panels (b–c) keep the DINOv3-S backbone fixed while replacing MSP with the structurally different DEIMv2 interface (Huang et al., 2025a), isolating the effect of detector topology. Figure 3 visualizes these architecture-specific realizations, while Table 1 summarizes the corresponding intervention magnitudes and destination- or carrier-level evidence.

Changing the backbone shifts the dominant destination. Holding the MSP detector interface fixed, replacing DINOv3-S with native DINOv2-S/14 shifts the dominant deep-semantic dependency from $F_{11} \to P_4$ to $F_{11} \to P_5$, as illustrated in Figure 3(a–b). On Full COCO, zeroing $F_{11} \to P_5$ causes a 21.01 AP drop, whereas every shallower branch feeding P_5 contributes at most 0.21 AP—roughly two orders of magnitude smaller. Cross-image permutation further increases the

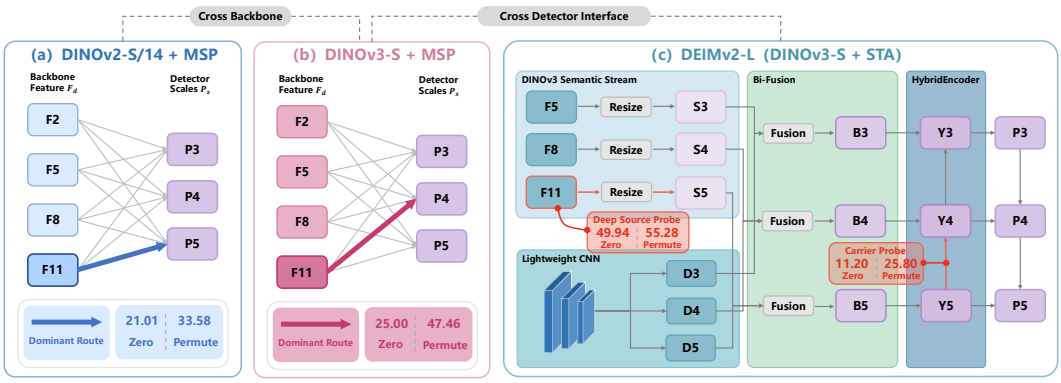

Figure 3: **System-dependent spatial realization of deep-semantic dependence. (a–b)** With MSP fixed, changing the backbone shifts the dominant route from $F_{11} \to P_5$ to $F_{11} \to P_4$. **(b–c)** With DINOv3-S fixed, changing the detector interface reroutes the same dependence through a P_5-centered carrier.

degradation to 33.58 AP, and decoder-level masking likewise identifies P_5 as the spatial memory with the largest aggregate dependence. Thus, even with detector topology held fixed, changing the backbone representation is sufficient to reshape the dominant spatial destination.

Table 1: **Cross-system interventional audit.** The dominant deep-semantic dependency changes its spatial realization across backbones and detector interfaces.

System	Dominant dependency	Zero	Perm.	Supporting evidence
DINOv3-S + MSP	$F_{11} \to P_4$	25.00	47.46	P_4 memory masking: 31.22 AP drop
DINOv2-S/14 + MSP	$F_{11} \to P_5$	21.01	33.58	P_5: largest memory dependence
DEIMv2-L	Deepest semantic source	49.94	55.28	$P_5 \to P_4$ carrier: 11.20 / 25.80 AP drop (zero / perm.)

Changing detector topology changes the spatial carrier. We next hold the DINOv3-S backbone fixed and replace MSP with the DEIMv2 interface. The deepest semantic source remains overwhelmingly important: zeroing and cross-image permutation cause drops of 49.94 and 55.28 AP, respectively. Its spatial realization, however, differs qualitatively from the direct MSP routing pattern. DEIMv2 propagates deep semantics through a P_5-centered internal hierarchy, including a top-down $P_5 \to P_4$ carrier. Intervening on this carrier causes an 11.20 AP drop under zeroing and a 25.80 AP drop under cross-image permutation (Table 1), showing that the carrier itself conveys image-specific information. Thus, the importance of deep semantics persists, while the path through which that information reaches the detector is reorganized by the interface topology.

Across the three systems, strong image-specific deep-semantic dependence recurs, but its spatial realization is system-dependent. We therefore use the dependency map as a design prior: strong paths should be preserved, weak source branches motivate targeted pruning, and carrier-mediated dependencies motivate joint topology–depth redesign. Section 4 tests these hypotheses through controlled retraining.

4 From Diagnosis to Coupled Interface Design

4.1 Why Weak Marginal Dependence Does Not Imply Structural Redundancy

The DINOv3-S + MSP diagnosis in Section 3.2 exposes two signals that appear, at first glance, to point in different directions. On the one hand, the decoder exhibits only weak marginal dependence on the P_5 spatial memory. On the other hand, deep semantic information is strongly concentrated along the $F_{11} \to P_4$ route. If weak endpoint dependence directly implied structural redundancy,

a natural diagnosis-derived hypothesis would be to remove P_5 while leaving the remainder of the interface unchanged.

We test this hypothesis by retraining a simplified P_3/P_4 interface under the same optimization protocol as the original $P_3/P_4/P_5$ baseline. This isolated-pruning variant, denoted M2, performs worse than the baseline M0 by 1.13 Raw AP and 0.78 EMA AP. Thus, the weak marginal dependence observed in the fixed checkpoint does not translate into independent retraining-time redundancy. Removing a spatial endpoint changes the topology within which the remaining depth features are optimized and used; endpoint weakness is therefore better interpreted as a signal for coordinated topology restructuring than as evidence that the endpoint can be deleted in isolation.

This failure raises whether topology simplification changes the value of deep-semantic emphasis. To test this question in a controlled setting, we reuse a fixed F_{11}-heavy depth treatment from earlier architecture exploration and treat it as the second factorial factor. The treatment assigns greater weight to the deepest feature when constructing P_4, remains fixed throughout training, and introduces no additional learnable routing parameters. Appendix D documents its provenance and an independent fixed-versus-learnable gate audit.

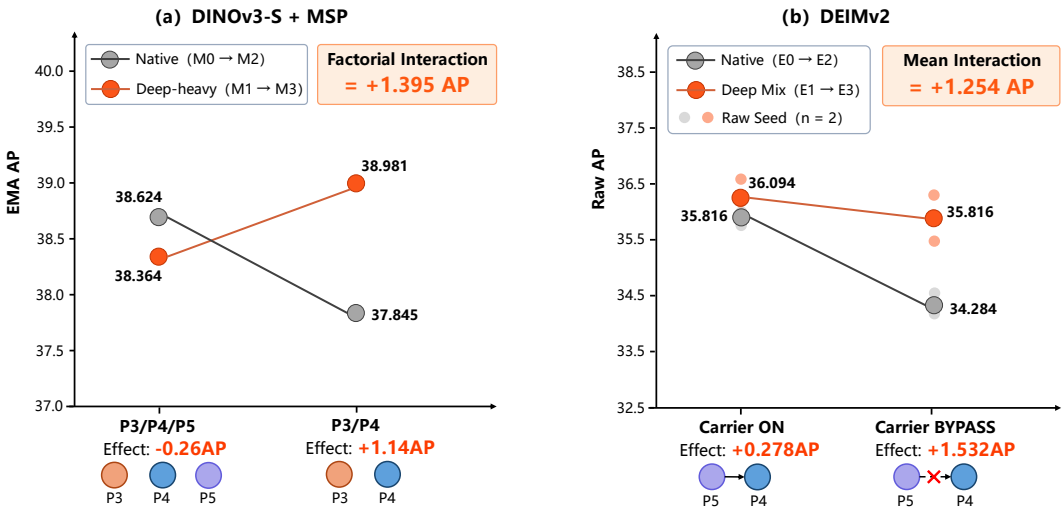

Figure 4: **Factorial evidence for topology-dependent deep-semantic treatment. (a)** In DINOv3-S + MSP, the deep-heavy treatment reverses from detrimental to beneficial after simplifying the spatial-memory topology. **(b)** In DEIMv2, Deep Mix becomes substantially more valuable after bypassing the native $P_5 \to P_4$ carrier. Both experiments reveal topology-dependent treatment effects.

4.2 Factorial Evidence for Depth–Topology Coupling

We next test whether the effect of depth treatment depends on spatial-memory topology using a 2×2 factorial design over two controlled factors: spatial-memory topology ($P_3/P_4/P_5$ versus P_3/P_4) and depth treatment (native fusion versus the fixed deep-heavy prior introduced in Section 4.1). This yields four configurations—M0 through M3—which are trained under the same optimization protocol with all non-factorial hyperparameters held fixed.

A clear crossover interaction. Under the full $P_3/P_4/P_5$ topology, applying the deep-heavy treatment slightly reduces performance: M1 is lower than M0 by 0.49 Raw AP and 0.26 EMA AP. Under the simplified P_3/P_4 topology, however, the same treatment has the opposite effect: M3 exceeds M2 by 1.38 Raw AP and 1.14 EMA AP. The treatment effect reverses sign when the spatial-memory topology changes, as shown in Figure 4(a).

We quantify the modulation by $\mathcal{I} = (\mathrm{M3} - \mathrm{M2}) - (\mathrm{M1} - \mathrm{M0})$, which yields $+1.395$ AP for EMA evaluation and $+1.864$ AP for the raw checkpoints. The crossover establishes depth treatment and spatial-memory topology as coupled design variables: the same deep-heavy treatment is slightly harmful under $P_3/P_4/P_5$ but beneficial under P_3/P_4. Thus, weak P_5 dependence is better inter-

preted as an opportunity for coordinated restructuring than as evidence of independent removability. An independent M0/M3 replication further supports the final design contrast ($+0.600$ EMA AP; $+0.718$ Raw AP).

4.3 Cross-Architecture Validation of Functional Compensation

To test whether this topology dependence generalizes beyond MSP, we perform a second 2×2 factorial crossing semantic-source treatment (Native versus Deep Mix) with carrier topology (ON versus BYPASS). Native uses F_8 as the initial P_4 semantic source, whereas Deep Mix replaces it with the fixed mixture $0.238F_8 + 0.762F_{11}$. BYPASS skips the first $P_5 \to P_4$ top-down fusion and forwards the current pre-fusion P_4 tensor directly to the remaining encoder stages.

Functional compensation across architectures. With the native $P_5 \to P_4$ carrier intact, Deep Mix provides only a modest two-seed mean gain of $+0.278$ Raw AP. Carrier bypass substantially weakens the native model, but under this topology Deep Mix becomes much more valuable, improving E3 over E2 by $+1.532$ AP and restoring the descriptive mean to the E0 level (Figure 4(b)). The factorial interaction is $+1.254$ AP and remains positive in both seeds ($+1.239$ and $+1.268$ AP). Together with the DINOv3-S + MSP crossover, this provides cross-architecture evidence that the value of direct deep-semantic access depends on the spatial topology through which that information is transported.

4.4 Diagnosis-Guided Targeted Pruning on DINOv2

The DINOv2 diagnosis suggests a qualitatively different redesign strategy from the DINOv3 case. In DINOv3-S + MSP, the weak diagnostic signal appears at an entire spatial endpoint, motivating topology-level restructuring. In DINOv2-S/14 + MSP, the P_5 memory exhibits strong decoder-level dependence, while its internal source contributions are highly uneven: $F_{11} \to P_5$ is the dominant route, whereas the shallower $F_2 \to P_5$ and $F_5 \to P_5$ branches have negligible marginal effects. The resulting design signal is more local: preserve the P_5 endpoint, but simplify its source set.

Following this diagnosis, we construct a targeted pruning variant that reduces the P_5 source set from $\{F_2, F_5, F_8, F_{11}\}$ to $\{F_8, F_{11}\}$, while leaving the P_3/P_4 topology and the rest of the training protocol unchanged. On COCO-Medium, this physically pruned model preserves detection accuracy while substantially reducing the parameterization associated with the P_5 projection path, providing a direct test of the source-level redundancy identified by the dependency map. Full-COCO validation of this design is reported in Section 5.

The contrast between DINOv3 and DINOv2 reveals that diagnosis identifies not only which dependencies matter, but also the structural granularity at which redesign should act. DINOv3 requires coordinated restructuring at the endpoint/topology level, whereas DINOv2 supports localized source-level pruning without changing the surrounding spatial topology.

5 Full-Scale Validation and Efficiency

We evaluate the final DINOv3 M3 and DINOv2 V1 designs on COCO 2017 (Lin et al., 2014). Within each backbone family, the baseline and redesign use matched optimization settings, the same training seed, and identical 30-epoch schedules; final-epoch EMA is the primary endpoint. Raw checkpoints and full metric breakdowns are provided in Appendix F.

5.1 Full-COCO Confirmation of Diagnosis-Guided Designs

Table 2 reports the Full-COCO evaluation of the final diagnosis-guided designs, comparing the DINOv3 coupled redesign M3 against its matched baseline M0 and the DINOv2 targeted-pruning variant V1 against its matched baseline V0.

DINOv3 coupled redesign. Under the matched seed-43 comparison, M3 reaches 53.658 AP, compared with 53.589 AP for M0, corresponding to a $+0.069$ AP difference. The coupled redesign coupled redesign therefore preserves aggregate Full-COCO accuracy after complete retraining. We additionally perform a paired image-level bootstrap (Efron & Tibshirani, 1993) on the matched seed-43 checkpoints. The observed $+0.069$ AP difference yields a 95% percentile interval of

Table 2: **Full-COCO confirmation of the final diagnosis-guided designs.** Primary comparisons use matched seeds and final-epoch EMA checkpoints.

Backbone	Variant	Seed	AP	Params (M)	GFLOPs
DINOv3-S	M0 (P_{3+4+5})	43	53.589	40.57	159.52
DINOv3-S	M3 (P_{3+4})	43	53.658	33.76 (-16.78%)	153.70 (-3.65%)
DINOv2-S/14	V0 (P_5 :4-src)	42	51.416	43.03	158.15
DINOv2-S/14	V1 (P_5 :2-src)	42	51.300	40.18 (-6.63%)	155.87 (-1.44%)

$[-0.205, +0.443]$, which includes zero. Accordingly, we interpret the Full-COCO result as aggregate accuracy preservation rather than as evidence of a resolved accuracy improvement. Complete metric-wise intervals and the resampling protocol are provided in Appendix G.

DINOv2 targeted pruning. For DINOv2-S/14, the matched seed-42 comparison gives 51.300 AP for V1 versus 51.416 AP for V0, a difference of -0.117 AP. This near-parity result is consistent with the source-level diagnosis in Section 4.4: the critical P_5 endpoint and its dominant deep-semantic route are retained, while weak shallow-source branches are removed. The scale-specific metrics reveal a structured trade-off rather than a uniform change across object sizes. AP_M is essentially preserved, increasing by $+0.055$ AP, while AP_S and AP_L decrease by 0.864 and 0.639 AP, respectively. Full AP_{50}, AP_{75}, and scale-specific results are reported in Appendix F.

5.2 Efficiency Profile

The two diagnosis-guided redesigns reduce complexity at different structural levels, consistent with the granularity exposed by the dependency analysis. For DINOv3, M3 removes the P_5 memory and its associated projector/decoder capacity, reducing total parameters by 16.78%, interface parameters by 64.06%, and estimated GFLOPs by 3.65%. For DINOv2, V1 preserves the overall spatial topology and removes only the weak P_5 source branches, reducing total parameters by 6.63%, P_5-projector parameters by 42.22%, and estimated GFLOPs by 1.44%. These reductions mirror the diagnosis itself: DINOv3 supports topology-level restructuring, whereas DINOv2 supports localized intra-memory simplification. Supporting eager-mode measurements on RTX 5090 also favor M3 (28.2→22.8 ms FP32; 35.0→28.8 ms AMP); the full latency protocol is given in Appendix F.

6 Conclusion and Limitations

We introduced Diagnose-Before-Design, a diagnosis-guided workflow for analyzing and redesigning the depth-to-scale interface between plain foundation ViTs and multiscale object detectors. By perturbing existing depth-to-scale paths and spatial memories without retraining, we directly expose how trained detectors functionally use their interfaces. Across DINOv2, DINOv3, and distinct detector topologies, we find a consistent principle: strong image-specific deep-semantic dependence repeatedly emerges, but its spatial realization is system-dependent rather than governed by a universal depth-to-scale assignment. Controlled factorial experiments further show that depth treatment and spatial topology are coupled design variables, revealing why weak marginal dependence does not necessarily imply retraining-time redundancy. These diagnoses explain and validate topology–depth restructuring for DINOv3 and motivate targeted source pruning for DINOv2, both preserving near-parity Full-COCO accuracy while reducing interface capacity. More broadly, our results suggest that foundation-ViT detector interfaces should be measured before they are redesigned: functional dependency maps provide an empirical basis for deciding not only what to simplify, but also at what structural granularity.

Clarifying the scope of applicability of this study helps to further leverage the guiding value of the diagnosis. The current analysis focuses on DINO-family self-supervised ViT-S backbones and COCO object detection. Moreover, interventional dependence characterizes how a fixed trained model uses its internal paths and does not directly determine the dependency structure that emerges after retraining. Extending the analysis to larger backbones, other pretraining paradigms, and other dense-prediction tasks would further test the generality of the observed dependencies.

AI USE STATEMENT

Generative AI tools were used to assist with literature searches, research conceptualization, aspects of experimental design, and the drafting and revision of the manuscript. AI-generated suggestions served solely as a reference aid. All cited references were verified by the authors, and all matters concerning methodological choices, experiments, analysis, interpretation of results, and the final content of the manuscript were reviewed and approved by the authors. The authors bear full responsibility for the accuracy, originality, and completeness of this study.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce the reported analyses and comparisons throughout the main paper and appendix. Section 3.1 defines the intervention protocol and evaluation setting. Appendix A documents the construction and representativeness of COCO-Medium, while Appendix B provides implementation checks for the interventions. Complete dependency results, DINOv3 and DEIMv2 factorial configurations, Full-COCO endpoint metrics and efficiency measurements, and the paired bootstrap protocol are reported in Appendices C–G. Primary Full-COCO comparisons use matched training seeds and fixed evaluation endpoints as specified in Section 5.

Code and experiment configurations for the experiments reported in this paper are available at https://anonymous.4open.science/r/PC-DETR-Anonymous-CPC3. The repository and supplementary materials describe the experiment settings, data preparation, and evaluation procedures.

ACKNOWLEDGMENTS

Omitted for double-blind review.

REFERENCES

Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep ViT features as dense visual descriptors. *arXiv preprint arXiv:2112.05814*, 2021.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6541–6549, 2017.

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. GAN dissection: Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations*, 2019.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pp. 213–229, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Qiang Chen, Xiangbo Su, Xinyu Zhang, Jian Wang, Jiahui Chen, Yunpeng Shen, Chuchu Han, Ziliang Chen, Weixiang Xu, Fanrong Li, Shan Zhang, Kun Yao, Errui Ding, Gang Zhang, and Jingdong Wang. LW-DETR: A transformer replacement to YOLO for real-time detection. *arXiv preprint arXiv:2406.03459*, 2024.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *International Conference on Learning Representations*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Shihua Huang, Yongjie Hou, Longfei Liu, Xuanlong Yu, and Xi Shen. Real-time object detection meets DINOv3. *arXiv preprint arXiv:2509.20787*, 2025a.

Shihua Huang, Zhichao Lu, Xiaodong Cun, Yongjun Yu, Xiao Zhou, and Xi Shen. DEIM: DETR with improved matching for fast convergence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15162–15171, 2025b.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3519–3529. PMLR, 2019.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pp. 280–296, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Namuk Park, Wonjae Kim, Byeongho Heo, Taekyung Kim, and Sangdoo Yun. What do self-supervised vision transformers learn? In *International Conference on Learning Representations*, 2023.

Yansong Peng, Hebei Li, Peixi Wu, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. D-FINE: Redefine regression task of DETRs as fine-grained distribution refinement. In *International Conference on Learning Representations*, 2025.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Advances in Neural Information Processing Systems*, volume 34, pp. 12116–12128, 2021.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12179–12188, 2021.

Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. RF-DETR: Neural architecture search for real-time detection transformers. In *International Conference on Learning Representations*, 2026.

Byungseok Roh, JaeWoong Shin, Wuhyun Shin, and Saehoon Kim. Sparse DETR: Efficient end-to-end object detection with learnable sparsity. In *International Conference on Learning Representations*, 2022.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7262–7272, 2021. doi: 10.1109/ICCV48922.2021.00717.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16965–16974, 2024.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT pre-training with online tokenizer. In *International Conference on Learning Representations*, 2022.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

A Controlled Analysis Set and Evaluation Role

COCO-Medium is a frozen controlled analysis subset containing $5{,}000$ train2017 images and $2{,}000$ val2017 images. It is used for dense dependency scans, controlled factorial retraining, and audit experiments; key dependencies and final architecture endpoints are then confirmed with Full-COCO-trained checkpoints.

Category distribution. Train-set category frequencies correlate strongly with Full COCO (Pearson 0.999687; Spearman 0.9785; Jensen–Shannon divergence 0.000640). On the validation subset, Pearson correlation is 0.999515 and Jensen–Shannon divergence is 0.001127. The maximum category-frequency deviation is 0.520 percentage points on train and 0.394 points on validation.

Scale and box geometry. Small/medium/large object proportions differ from Full COCO by less than 0.2 percentage points. Across the audited continuous box-geometry variables, the maximum two-sample Kolmogorov–Smirnov distance is 0.0147.

Known sampling difference. COCO-Medium excludes the approximately 0.86%–0.96% empty-image fraction present in Full COCO, and therefore has a slightly higher number of annotated objects per image. We use it as a controlled mechanism-analysis set rather than as a replacement ranking benchmark.

B Intervention Protocol and Implementation Checks

Energy-matched branch intervention. For branch-zeroing controls, the target branch activation is removed and the remaining active branches feeding the same destination are rescaled so that the aggregate activation norm is matched. This separates semantic/path dependence from a trivial loss of activation magnitude.

Table 3: Uncompensated versus energy-matched branch zeroing for DINOv3-S + MSP on COCO-Medium. Values are AP drops from the baseline.

Backbone depth	$\mathbf{P_3}$ (Zero / Energy)	$\mathbf{P_4}$ (Zero / Energy)	$\mathbf{P_5}$ (Zero / Energy)
F_2	0.8645 / 0.8312	0.9414 / 0.9025	0.0237 / 0.0211
F_5	0.8406 / 0.8120	0.9195 / 0.8847	0.0143 / 0.0105
F_8	0.5389 / 0.5104	2.4676 / 2.4210	-0.0390 / -0.0412
F_{11}	0.8224 / 0.7955	32.6205 / 32.5841	0.2223 / 0.2180

The full matrix changes only marginally after energy matching, while the dominant $F_{11} \to P_4$ effect remains essentially unchanged (32.6205 versus 32.5841 AP).

Fixed-point-free batch permutation. For semantic-alignment tests, the intervened branch tensor is permuted across images within each inference batch while preserving tensor shape and the multiset of activation values. The permutation is constructed without fixed points so that no image receives its own target-branch feature. In the reported experiments, validation uses batch size 8.

Identity-hook verification. Before each intervention scan, an identity-hook audit verifies that registering the intervention hook without modifying activations reproduces the native checkpoint output. Across DINOv3-S + MSP, DEIMv2, and DINOv2-S/14 + MSP, the identity-hook evaluation matches the unhooked checkpoint AP to numerical precision.

Decoder-memory masking. Multiscale decoder memories are flattened and concatenated along the token dimension. Destination-level intervention masks the Key/Value tokens associated with one spatial memory immediately before decoder cross-attention, while leaving the upstream encoder representation and proposal-selection process unchanged.

C Complete Dependency Results

C.1 DINOv3-S + MSP: Complete Diagnosis Summary

Table 4: DINOv3-S + MSP dependency results across branch-, path-, and memory-level interventions.

Diagnostic level	Feature / intervention	$\mathbf{P_3}$	$\mathbf{P_4}$	$\mathbf{P_5}$
Source zeroing (Medium)	F_2	0.86	0.94	0.02
	F_5	0.84	0.92	0.01
	F_8	0.54	2.47	-0.04
	F_{11}	0.82	32.62	0.22
Permutation vs. zero	$F_{11} \to P_4$	–	32.62 / 38.31	–
Decoder masking (Medium)	Mask entire memory	8.71	36.77	0.74
Full-COCO confirmation	$F_{11} \to P_4$, zero / permute	–	25.00 / 47.46	–
	Mask entire memory	28.94	31.22	1.75

C.2 DINOv3-S + MSP: Full-COCO Scale-Specific Decomposition

Table 5: Full-COCO detection metrics under DINOv3-S + MSP interventions.

Condition	AP	$\mathbf{AP_{50}}$	$\mathbf{AP_{75}}$	$\mathbf{AP_S}$	$\mathbf{AP_M}$	$\mathbf{AP_L}$	$\mathbf{\Delta AP}$
Baseline (M0)	53.589	72.230	57.995	33.234	58.348	72.502	–
Zero $F_{11} \to P_4$	28.594	43.120	29.450	20.115	32.410	35.210	24.995
Permute $F_{11} \to P_4$	6.129	10.450	5.890	3.210	6.450	9.870	47.460
Mask P_3	24.651	38.410	25.120	2.210	27.890	45.670	28.938
Mask P_4	22.374	34.120	23.450	18.450	24.120	21.590	31.215
Mask P_5	51.843	70.890	56.120	33.120	57.890	68.450	1.746

D DINOv3 Diagnosis-to-Design Factorials

Topology $\times$ depth-treatment factorial (M0–M3). The depth-treatment effect changes from -0.488 Raw / -0.260 EMA AP under $P_3/P_4/P_5$ to $+1.376$ Raw / $+1.135$ EMA AP under P_3/P_4, yielding interaction terms of $+1.864$ Raw and $+1.395$ EMA AP. An independent M0/M3 replication gives $+0.718$ Raw and $+0.600$ EMA AP for M3 relative to M0; it is a replication of the final design contrast, not of the complete four-cell interaction.

Table 6: Complete M0–M3 factorial on COCO-Medium.

Cell	Configuration	Spatial topology	Depth treatment	Raw AP	EMA AP
M0	Baseline	$P_3/P_4/P_5$	Native fusion	38.771	38.624
M1	Deep prior	$P_3/P_4/P_5$	Fixed deep-heavy	38.283	38.364
M2	Isolated prune	P_3/P_4	Native fusion	37.642	37.845
M3	Coupled design	P_3/P_4	Fixed deep-heavy	39.017	38.981

Fixed deep-heavy treatment and provenance. The M0–M3 factorial uses the fixed gain vector

$$\mathbf{g} = [0.2486,\ 0.4130,\ 0.7943,\ 2.5441],$$

corresponding to normalized weights

$$\mathbf{p} = [0.0621,\ 0.1032,\ 0.1986,\ 0.6360]$$

over (F_2, F_5, F_8, F_{11}). This treatment predates the Diagnose-Before-Design study and originated during earlier architecture exploration of the DINOv3-based detector. Repository provenance records it as a strong F_{11}-heavy prior extracted from trained P_4 layer contributions, but the exact procedure that produced the numerical coefficients is not recoverable from the preserved code and experiment artifacts. We therefore do not interpret these coefficients as analytically derived, intervention-derived, or globally optimized.

The same deep-heavy shape was already present before the G0–G3 audit. G0–G3 compared this single strong prior against uniform weighting and tested fixed versus learnable scalar gates; it did not search over multiple non-uniform weighting schedules. The exact coefficients used by M1/M3 were frozen before the M0–M3 training runs. Consequently, the M0–M3 factorial should be interpreted as measuring the topology-dependent effect of this fixed pre-existing treatment, rather than as identifying an optimal depth-weighting rule.

Deep-prior and gate audit (G0–G3). This audit evaluates a pre-existing strong F_{11}-heavy prior against uniform weighting and separately tests whether scalar gate learnability is beneficial. It is not a search over alternative non-uniform prior shapes. In this interface skeleton, the fixed strong prior outperforms the fixed uniform treatment, while adding learnable gates provides no corresponding benefit. We therefore use the already specified fixed treatment as the depth factor in the subsequent M0–M3 factorial.

Table 7: Audit of depth prior and gate learnability on COCO-Medium.

Cell	Prior	Gate	Status	Raw best	EMA best	Raw last-5	EMA last-5
G0	Uniform	None	Fixed	38.001	38.088	37.499	37.993
G1	Uniform	Scalar	Learnable	37.936	38.084	37.418	37.837
G2	Deep-heavy	None	Fixed	38.764	38.578	38.173	38.391
G3	Deep-heavy	Scalar	Learnable	38.166	38.055	37.547	37.860

E DEIMv2 Cross-Architecture Factorial

We evaluate a 2×2 design over semantic-source treatment (Native versus Deep Mix) and carrier topology (Carrier ON versus BYPASS) using two independent seeds. Raw AP is the primary factorial statistic.

Deep Mix semantic-source treatment. Let F_5, F_8, and F_{11} denote the normalized DINOv3 patch-token features extracted from zero-based transformer block indices 5, 8, and 11, respectively, with the class token excluded. Before scale-specific processing, all three remain at the native stride-16 resolution. The native P_4 semantic source is

$$S_{P4}^{\text{native}} = F_8,$$

whereas Deep Mix replaces it with the fixed element-wise mixture

$$S_{P4}^{\text{deep}} = 0.238F_8 + 0.762F_{11}.$$

Table 8: Detailed DEIMv2 factorial results for seeds 44 and 45.

Cell	Source / carrier	Seed	Raw AP	EMA AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
E0	Native / Carrier ON	44	35.803	35.912	53.124	37.456	18.234	37.890	53.120
E0	Native / Carrier ON	45	35.829	35.885	53.089	37.512	18.190	37.912	53.180
E1	Deep Mix / Carrier ON	44	35.603	35.745	52.890	37.120	17.950	37.610	53.010
E1	Deep Mix / Carrier ON	45	36.585	36.612	54.120	38.340	18.780	38.890	54.210
E2	Native / BYPASS	44	34.455	34.520	51.230	35.890	16.890	36.210	51.450
E2	Native / BYPASS	45	34.113	34.210	50.890	35.450	16.540	35.980	51.020
E3	Deep Mix / BYPASS	44	35.494	35.580	52.670	37.100	17.890	37.540	52.890
E3	Deep Mix / BYPASS	45	36.137	36.210	53.560	37.890	18.450	38.320	53.670

The coefficients are fixed and introduce no learnable mixing parameters or additional projection layers. The mixture is formed before the original P_4 projection path. The resulting S_{P4} is concatenated with the unchanged stride-16 STA detail feature and passed through the original 1×1 convolution and SyncBatchNorm before entering the HybridEncoder. The semantic sources of P_3 and P_5 remain F_5 and F_{11}, respectively; therefore, under Deep Mix, F_{11} contributes to both the P_4 and P_5 semantic streams. The DINOv3 feature extractor itself remains trainable.

Carrier ON versus BYPASS. The carrier factor acts only on the first top-down stage of the HybridEncoder, which transports information from P_5 to P_4. Let q_4 denote the current P_4 tensor before this fusion and q_5 the Transformer-encoded P_5 tensor. With the carrier enabled, the P_4 representation passed to the subsequent top-down stage is

$$\widetilde{q}_4^{\mathrm{ON}} = B_0(q_4 + U(L_0(q_5))) \,,$$

where L_0 denotes the first lateral convolution, $U(\cdot)$ denotes upsampling, and B_0 denotes the first FPN block; the configured fusion operator is element-wise summation. Under BYPASS,

$$\widetilde{q}_4^{\mathrm{BYPASS}} = q_4 .$$

Thus, BYPASS skips the first lateral projection of q_5, its upsampling and addition into q_4, and the corresponding first FPN block. The resulting q_4 is inserted directly into the internal feature list, after which the second $P_4 \to P_3$ top-down stage, the PAN pathway, and the detector decoder proceed normally. Importantly, in E3 the bypassed q_4 already contains the Deep Mix semantic source; BYPASS therefore removes the first $P_5 \to P_4$ carrier rather than reverting P_4 to the native F_8 source.

One implementation detail is that the P_5 carrier retained for subsequent PAN processing differs between the two paths: Carrier ON retains $L_0(q_5)$, whereas BYPASS retains the unprojected q_5. The P_5 Transformer encoding, subsequent PAN processing, and decoder-level P_5 endpoint remain active in both conditions. The bypassed lateral and FPN modules remain instantiated in the model but are not executed in the BYPASS forward path. Consequently, all four factorial cells have the same registered parameter count of $32{,}551{,}214$; BYPASS changes the executed computation graph rather than pruning parameters from the model.

F Full-COCO Endpoint Metrics and Efficiency

Table 9: Full-COCO endpoint metrics and deterministic complexity measures. Seed 47 is an additional M3 replication and is not a matched-seed baseline comparison.

System / design	Checkpoint	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params	Interface params	GFLOPs
DINOv3 M0	EMA seed 43	53.589	72.230	57.995	33.234	58.348	72.502	40.57M	10.63M	159.52
DINOv3 M3	EMA seed 43	53.658	72.346	57.637	32.669	58.161	73.275	33.76M	3.82M	153.70
DINOv3 M3 replication	EMA seed 47	53.753	72.438	57.892	32.755	58.451	72.787	33.76M	3.82M	153.70
DINOv2 V0	EMA seed 42	51.416	69.924	55.436	31.657	55.678	70.824	43.03M	10.63M	158.15
DINOv2 V1	EMA seed 42	51.300	69.889	55.110	30.793	55.733	70.185	40.18M	7.78M	155.87

PyTorch eager model-forward latency. Latency is measured on a single NVIDIA RTX 5090 with batch size 1 using a bidirectional test-order protocol. Each condition uses 200 warm-up iterations followed by 1,000 measured forward passes; the order is reversed in the second run to reduce runtime-order bias.

Table 10: Aggregate PyTorch eager model-forward latency. Absolute latency is runtime- and platform-sensitive; parameter count and GFLOPs are the primary deterministic efficiency measures.

Precision	M0 latency (ms)	M3 latency (ms)	Difference
FP32	28.2	22.8	-5.4
AMP	35.0	28.8	-6.2

G Paired Image-Level Bootstrap Uncertainty Analysis

For each bootstrap replicate, $5{,}000$ COCO val2017 image IDs are sampled with replacement. Both evaluated checkpoints are scored on the same resampled image set and the paired metric difference Δ is recorded. We use $1{,}000$ replicates and report percentile intervals. This analysis quantifies validation-image sampling uncertainty for fixed checkpoints; it does not include training-seed uncertainty.

For overall AP, the percentile interval includes zero for both M3 runs. We therefore use the bootstrap as an uncertainty characterization around the observed fixed-checkpoint differences, while the Full-COCO endpoint claim is accuracy preservation with reduced interface capacity.

Table 11: Paired image-level bootstrap results for DINOv3 M3 versus the fixed M0 reference on Full COCO.

Training seed	Metric	Observed	Bootstrap mean	95% percentile interval	Empirical $P(\Delta > 0)$
43	AP	+0.0691	+0.0684	$[-0.2051, +0.4427]$	70.6%
43	AP_{50}	+0.1164	+0.1158	$[-0.2815, +0.5808]$	74.2%
43	AP_{75}	-0.3585	-0.3591	$[-0.7485, +0.1282]$	8.4%
43	AP_S	-0.5652	-0.5620	$[-1.6032, +0.8681]$	24.1%
43	AP_M	-0.1866	-0.1842	$[-0.7313, +0.3879]$	26.8%
43	AP_L	+0.7733	+0.7712	$[-0.0254, +1.5837]$	96.6%
47	AP	+0.1642	+0.1499	$[-0.2105, +0.4669]$	82.6%
47	AP_{50}	+0.2078	+0.2015	$[-0.2328, +0.6377]$	84.2%
47	AP_{75}	-0.1027	-0.1442	$[-0.5620, +0.2639]$	25.0%
47	AP_S	-0.4796	-0.3155	$[-1.4716, +0.8294]$	30.3%
47	AP_M	+0.1034	+0.1225	$[-0.2942, +0.5441]$	72.5%
47	AP_L	+0.2854	+0.3396	$[-0.6225, +1.2936]$	75.3%