Sparse All-Layer Connector for Domain Generalised Semantic Segmentation

Ruoyu Guo* **Xin Kun Lin*** **Maurice Pagnucco** **Yang Song**
School of Computer Science and Engineering, University of New South Wales
{ruoyu.guo, xin_kun.lin, morri, yang.song1}@unsw.edu.au

Abstract

Domain generalised semantic segmentation (DGSS) has recently benefited from combining multiple foundation models, such as vision-language and vision foundation models, to leverage their diverse representations for generalisation. Existing multi-model approaches share a common design choice: cross-model fusion is restricted to depth-aligned layers, implicitly assuming that mutually beneficial information resides at the same depth across models. However, foundation models pretrained under different objectives may develop distinct representation hierarchies, making the optimality of depth-aligned fusion questionable. Moreover, we observe that the foundation models exhibit different depth-wise domain sensitivity. Motivated by these observations, we propose a Sparse All-Layer Connector (SALC) that enables each layer to access information from preceding layers of another model. As the number of accessible layers grows with depth, dense aggregation may hurt generalisation. SALC therefore learns to adaptively select a sparse subset of informative layers. We further introduce a candidate dropout regularisation that strengthens sparsity and encourages SALC to explore diverse layers, leading to more robust selection. Across four foundation model combinations and three DGSS evaluation settings, the proposed design demonstrates stronger generalisation than conventional depth-aligned fusion. The code is available at https://github.com/RuoyuGuo/SALC.

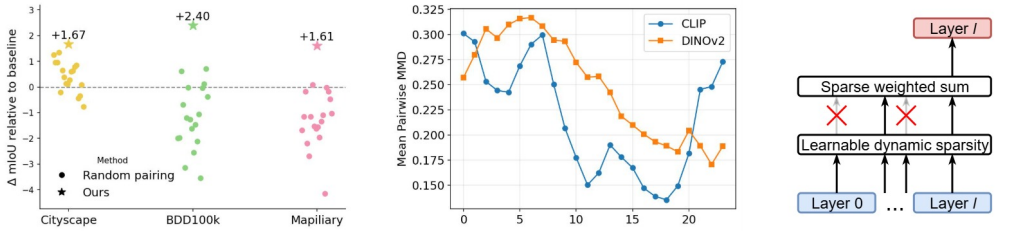
1 Introduction

Domain generalised semantic segmentation (DGSS) seeks to learn a segmentation model that performs robustly on arbitrary unseen target domains using only source domain data. To extract robust features, prior work has explored a range of strategies [14, 47], while more recent work leverages foundation models for their strong generalisability [60]. Despite these advances, relying on a single foundation model limits the diversity of representations, as each model is bounded by its pretrained knowledge. This has motivated a growing body of work that integrates multiple foundation models to learn heterogeneous representations for DGSS, such as vision-language models (VLMs), depth estimation foundation models (DFMs), and vision foundation models (VFMs) [71, 11, 12, 45, 63].

In practice, existing multi-model approaches [71, 11, 12] adopt depth-aligned fusion as a natural default: a learnable connector (e.g., linear or cross-attention layers) is placed at the l-th layer of one foundation model to fuse features from the l-th layer of the other. Previous studies have found that this approach can fuse invariant features or visually grounded features into a stronger representation [11]. However, models pretrained under different objectives may process information along distinct hierarchies [50, 56] and prior findings have shown that foundation models exhibit different transferability across layers [55, 43]. In such cases, it is unclear whether depth-aligned pairing remains the most effective choice.

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Random layer pairing w.r.t. the depth-aligned baseline (b) Overall cross-domain MMD across depths (c) Sparse all-layer connector (Ours)

Figure 1: (a) The model is trained on GTAV and evaluated on Cityscapes, BDD100K, and Mapillary. The random pairing is sampled independently for DINOv2 and CLIP using over 40 seeds. (b) Layer-wise cross-domain behaviour measured by MMD. We report the average pairwise MMD over all six domain pairs of the four domains. (c) Architecture of our sparse all-layer fusion.

To examine this empirically, we study a representative combination [24, 71, 59], a VLM (CLIP [49]) and a VFM (DINOv2 [44]). We compare the depth-aligned baseline with a variant that randomly pairs layers across the two models. As shown in Figure 1 (a), random pairing yields mixed results relative to depth-aligned fusion, suggesting that the effectiveness is sensitive to layer matching. We further analyse layer-wise cross-domain behaviour using Maximum Mean Discrepancy (MMD). Figure 1 (b) shows that CLIP and DINOv2 exhibit markedly different depth-wise patterns: CLIP reaches its lowest at middle layers and increases toward the end, while DINOv2 decreases monotonically with depth. These observations motivate us to fuse features between models in a flexible cross-depth manner. In this paper, for each layer in one model, which we refer to as a *query layer*, we allow it to aggregate information from multiple layers of the other model, which we refer to as *candidate layers*.

However, this flexible design poses two critical challenges. First, the number of candidate layers grows with depth, and the informativeness of each candidate layer varies across tokens, which complicates effective fusion. Second, dense aggregation across all candidate layers can introduce irrelevant or domain-specific features, which may hurt generalisation. Although some existing methods [6, 53] propose various multi-modal fusion strategies, they typically utilise cross-attention and TopK routing, which are insufficient here. This is because cross-attention uses softmax, which assigns non-zero weights to every candidate layer and may retain irrelevant features, while TopK routing enforces sparsity but with a fixed K that cannot adapt across depths and tokens.

To this end, we propose the **Sparse All-Layer Connector (SALC)** for cross-model fusion in DGSS. Different from existing methods that restrict fusion to depth-aligned layers, SALC allows each query layer to access an adaptive subset of all preceding layers up to the same depth in the other model, so that only the most relevant features are fused. To learn adaptive selection, we adopt Sparsegen [31], which projects predicted layer scores into a sparse weight distribution over candidate layers. Unlike softmax or TopK routing, Sparsegen can control the active candidate layers through a sparsity factor rather than a manually defined number. Building on this, we predict a token-wise sparsity factor using the candidate layer at the same depth, thereby enabling adaptivity at both the *token* and *layer* levels.

Moreover, we propose a candidate dropout regularisation that perturbs over-selected layers during training, guiding SALC toward more robust and discriminative selection. We explore two instantiations: *uniform drop*, where the same drop rate is applied for all layers and *frequency-based drop*, where a higher drop rate is applied for over-used layers. We evaluate SALC on five foundation models spanning VLMs (CLIP, SigLIP [70]), VFMs (DINOv2, SAM [29]), and a depth model (DepthAnythingV2 [64]) across three evaluation settings, showing competitive performance compared to depth-aligned fusion. Our main contributions are summarised as follows:

- We propose SALC, a multi-model DGSS framework that aggregates cross-layer features from different foundation models through a sparse selection mechanism. The selection is adaptively controlled for each layer and token through a learnable sparsity factor.
- To achieve more robust layer selection, we further introduce a candidate dropout regularisation that strengthens sparsity and exploration of informative layers.
- We demonstrate the effectiveness of SALC and provide an in-depth analysis of how cross-layer selection benefits domain generalisation. We also achieve competitive performance on three types of foundation models across three DGSS evaluation settings.

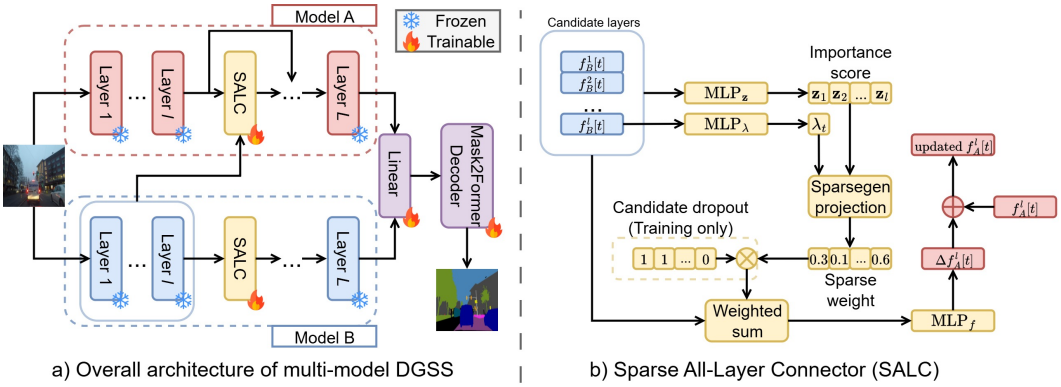


Figure 2: (a) Two models extract features from the same image in parallel. SALC fuses multi-layer features from one model into the other. The final segmentation map is predicted using multi-scale features of two models. (b) SALC selectively fuses features from all candidate layers, controlled by a learnable sparsity factor. We also propose a dropout regularisation to improve robustness.

2 Related work

Domain generalised semantic segmentation (DGSS). DGSS requires the segmentation model to remain robust when encountering unseen images [16, 18, 27, 33, 38]. Classic approaches mainly follow three directions, including data augmentation [68, 32, 28, 74, 19, 7, 25, 23, 2, 17, 73, 61], learning invariant representation [1, 20, 62, 19, 14, 5] and feature normalisation [46, 47, 48, 9]. Despite their effectiveness, these methods rely on models trained with less data. Recent methods adapt foundation models to DGSS through parameter-efficient fine-tuning [60, 21] or selectively tuning domain-sensitive parameters and features [4, 69, 66, 72, 3]. To extract more diverse cues, several methods fuse features from multiple foundation models, such as language [45, 71, 42, 22, 36], depth map [12, 11], and high-resolution images [57]. Unlike prior multi-model DGSS methods that primarily explore different model combinations, we improve generalisation through the fusion design itself. We propose flexible cross-depth layer fusion together with a frequency-based dropout regularisation that encourages SALC to discover more robust features.

Multi-model feature fusion. Multi-model feature fusion aims to exploit complementary representations from different encoders or modalities [35]. In multimodal large language models, one line of work enhances visual perception by combining visual tokens from multiple encoders, resolutions, or expert branches [37, 26, 58, 34, 53, 54, 35]. Another line aggregates multi-layer features before feeding into the language model with depth-aligned or dense connectors [39, 40, 6, 24, 8, 10, 65]. In DGSS, recent multi-model methods fuse complementary cues from VFMs, VLMs, and depth foundation models, but their cross-model interactions are typically established between depth-aligned layers [71, 11, 12]. Unlike these works, SALC tackles cross-model layer assignment as a learnable problem: for each query layer, it adaptively selects a sparse subset of candidate layers.

3 Method

3.1 Overview

Problem formulation. We address the DGSS problem, where a model is trained on a labelled source domain $\mathcal{D}_s$ and evaluated on unseen target domains $\mathcal{D}_t$ whose samples are inaccessible during training. We use $\mathcal{D}_s \rightarrow \mathcal{D}_t$ to denote training on $\mathcal{D}_s$ and evaluating on $\mathcal{D}_t$.

Overall architecture. As illustrated in Figure 2, our framework consists of two frozen encoders $\mathcal{E}_A$ and $\mathcal{E}_B$ pretrained under different objectives and a segmentation decoder. The core component of our framework is a set of SALCs attached after each layer, allowing each *query layer* to access all preceding *candidate layers* while activating only a sparse subset for aggregation. To improve the robustness of selection, we further propose a frequency-based regularisation. Given an input RGB image, the two encoders extract features in parallel. The last-layer features from both encoders are

then concatenated and projected by a linear layer into a unified representation, which is fed into a Mask2Former [13] decoder to produce the final segmentation map.

3.2 Sparse all-layer connector

Revisiting depth-aligned fusion. In prior multi-model DGSS works [71, 11, 12], given an input image x, it is fed into two encoders $\mathcal{E}_A$ and $\mathcal{E}_B$ with L transformer layers each. We denote the output of their l-th layer as $f_A^l, f_B^l \in \mathbb{R}^{T\times D}$, where T and D are the number of tokens and feature dimensions. A depth-aligned connector $\mathcal{C}_l$ is employed to fuse features from the same layer of each encoder and residually updates the query encoder: $f_A^l \leftarrow f_A^l + \Delta f_A^l$, where $\Delta f_A^l = \mathcal{C}_l(f_A^l, f_B^l) \in \mathbb{R}^{T\times D}$. Such depth-aligned pairing, however, is not necessarily optimal.

Here, we propose SALC that jointly learns candidate weights and adaptively controls the number of active candidate layers. Unlike prior work [71, 63, 11], SALC produces an update signal $\Delta f_A^l \in \mathbb{R}^{T\times D}$ that aggregates information from *all* accessible candidates $\{f_B^1, \ldots, f_B^l\}$ rather than f_B^l alone. Moreover, it learns token-specific sparsity levels for each layer. Let $[t]$ denote indexing at the t-th token position, we describe how $\Delta f_A^l[t]$ is computed in three steps: (1) predicting importance scores for each candidate layer, (2) projecting them into a sparse weight distribution, and (3) aggregating candidates with the resulting weights. This pipeline is applied independently at each token position and symmetrically to $\mathcal{E}_B$.

Importance scoring. For the l-th query layer, we collect the outputs at position t from all accessible candidate layers as $\mathbf{F}_t = [f_B^1[t], \ldots, f_B^l[t]] \in \mathbb{R}^{l\times D}$. We then predict importance scores for l layers using an MLP, while an RMSNorm is applied first to mitigate the magnitude variations across depths:

$$\mathbf{z}_t = \mathrm{MLP}_z\big(\mathrm{RMSNorm}(\mathbf{F}_t)\big) \in \mathbb{R}^l, \tag{1}$$

Sparse weight projection. To adaptively select useful layers from $\mathbf{F}_t$ for aggregation, we leverage Sparsegen [31] to map $\mathbf{z}_t$ into a sparse weight distribution $\mathbf{w}_t \in \mathbb{R}^l$. Specifically, Sparsegen activates only the candidate layers whose importance scores exceed a threshold $\tau_t \in \mathbb{R}$. The threshold is jointly determined by $\mathbf{z}_t$ and a sparsity factor $\lambda_t < 1$. Intuitively, a larger λ_t results in a higher threshold, leading SALC to concentrate on fewer layers, while a smaller λ_t encourages SALC to aggregate more candidate layers. Formally, the sparse weights are computed as:

$$\mathbf{w}_t = \max\left(\frac{\mathbf{z}_t - \tau_t}{1-\lambda_t}, 0\right), \quad \text{where} \quad \tau_t = \frac{\sum_{k\in\mathcal{S}_t} \mathbf{z}_t[k] - 1 + \lambda_t}{|\mathcal{S}_t|}. \tag{2}$$

The active set $\mathcal{S}_t \subseteq \{1, ..., l\}$ contains the top-ranked candidate layers by their importance scores (we provide the procedure for computing the size of $\mathcal{S}_t$ in Appendix A).

Since semantic segmentation requires per-pixel predictions and spatially distinct tokens may benefit from different sparsity levels, we predict token-wise λ_t using f_B^l. This is because f_B^l provides a summarised view of all preceding candidate layers through residual updates, offering an effective signal to estimate the informativeness of previous features and determine the sparsity. Predicting λ_t from f_B^l also makes the sparsity input-dependent, allowing λ_t to vary with the content of each image rather than being fixed across inputs. We use an MLP for prediction:

$$\lambda_t = \mathrm{MLP}_\lambda(f_B^l[t]). \tag{3}$$

Candidate aggregation. We then aggregate $\mathbf{F}_t$ using the sparse weights $\mathbf{w}_t$, followed by an MLP that projects the result into the feature space of $\mathcal{E}_A$:

$$\Delta f_A^l[t] = \mathrm{MLP}_f\Big(\mathbf{w}_t^\top \cdot \mathbf{F}_t\Big) \in \mathbb{R}^D. \tag{4}$$

We use shared MLP_z, MLP_λ and MLP_f for all tokens in each SALC to reduce complexity. Across query layers, SALCs are independently parameterised and do not share weights.

3.3 Candidate dropout regularisation

The effectiveness of SALC relies on its ability to make input-dependent selection of candidate layers. However, during training, we observe that at certain query layers, the selection weights spread too

broadly across candidate layers, suggesting the module fails to make decisive choices at these layers. We thus propose a regularisation that drops accessible candidate layers, encouraging more decisive selection at every layer. We explore two instantiations: *uniform dropout* and *frequency-based dropout*. The regularisation is applied only during training and does not incur inference costs.

At each training step, we sample a binary mask $\mathbf{M} \in \{0,1\}^l$, where $\mathbf{M}[i] = 0$ $(1 \leq i \leq l)$ indicates dropping the i-th candidate. We stack the per-token weights from Equation (2) into $\mathbf{W} \in \mathbb{R}^{T \times l}$, multiply $\mathbf{W}$ by $\mathbf{M}$ to drop candidate layers, and renormalise each row so that the weights sum to 1 for every token. The renormalised $\tilde{\mathbf{W}}[i] \in \mathbb{R}^l$ then replaces $\mathbf{w}$ in Equation (4) for aggregation.

For sampling $\mathbf{M}$, we first explore *uniform dropout*, where each candidate layer is independently dropped with a fixed probability p. This approach treats all candidate layers equally. We further explore *frequency-based dropout*, where the drop probability p of each candidate layer is scaled by its historical selection frequency, giving dominant candidate layers a higher chance of being dropped. Specifically, we maintain two running statistics via exponential moving average (EMA). The first, $\bar{\mathbf{u}} \in \mathbb{R}^l$, tracks the average weight assigned to l candidate layers, obtained by averaging $\mathbf{W}$ over the token dimension. The second, $\bar{N} \in \mathbb{R}$, tracks the average number of active candidates (whose $\mathbf{W} > 0$) per token. We set the target usage to $\rho = 1/\bar{N}$. The drop probability for candidate layer i is:

$$p_i = p_{\max} \cdot \left[\max\left(\frac{\bar{\mathbf{u}}_i - \rho}{1 - \rho}, 0 \right) \right]^{\gamma}, \tag{5}$$

where $p_{\max}$ caps the drop rate and γ controls the penalty sharpness. Mask entries are sampled as $\mathbf{M}[i] \sim \text{Bernoulli}(1 - p_i)$ in both instantiations.

Intuitively, *frequency-based dropout* penalises only the candidates whose usage exceeds the target ρ, such that candidates with usage below ρ receive zero drop probability. With $\gamma > 1$, the drop rate grows sharply with overuse, concentrating the regularisation on the most over-used candidates. We apply this regularisation independently within each SALC, with $\bar{\mathbf{u}}$ and $\bar{N}$ tracked per-SALC to capture the selection pattern of each query layer.

3.4 Training objectives

Following prior multi-model DGSS work [71], we train SALC using the standard segmentation losses from Mask2Former [13]: $\mathcal{L}_{total} = \alpha_{bce}\mathcal{L}_{bce} + \alpha_{dice}\mathcal{L}_{dice} + \alpha_{cls}\mathcal{L}_{cls}$, where $\mathcal{L}_{bce}, \mathcal{L}_{dice},$ and $\mathcal{L}_{cls}$ represent the binary cross-entropy loss, the dice loss for the predicted masks, and the cross-entropy loss for each queried proposal, respectively.

4 Experiments

4.1 Datasets & evaluation protocols

Datasets. We evaluate SALC on five driving-scene semantic segmentation datasets that share 19 common classes. Specifically, GTAV (G) [51] and SYNTHIA (S) [52] are two synthetic datasets, containing 24,966 and 9,400 training images, respectively. Cityscapes (C) [15] consists of 2,975 training and 500 validation images. BDD100K (B) [67] and Mapillary (M) [41] include 1,000 and 2,000 validation images, respectively.

Evaluation. Following the evaluation protocols of existing DGSS methods [60, 4, 46], we evaluate SALC on three evaluation settings with the mIoU metric: (1) G→{C, B, M}, (2) S→{C, B, M}, and (3) C→{B, M}.

Implementation details. Following existing multi-model DGSS methods [71, 12, 11], we evaluate SALC on two VFMs (SAM [29], DINOv2 [44]), two VLMs (CLIP [49], SigLIP [70]), and one DFM (DepthAnythingV2 [64]). This results in four combinations of $\mathcal{E}_A$ and $\mathcal{E}_B$: DINOv2-CLIP, DINOv2-SigLIP, SAM-CLIP, and DINOv2-DepthAnythingV2. To ensure fair comparison, for the first three combinations, we use the framework of MFuser [71]. For DINOv2-DepthAnythingV2, we use the framework of DepthForge [12]. More details are shown in Appendix B.

Table 1: Comparison with depth-aligned fusion baselines (mIoU).

Backbone	Method	GTA5	SYNTHIA	Cityscapes
DINOv2-CLIP	MFuser	67.82	50.34	71.84
	Ours	**69.40**	**52.99**	**72.51**
DINOv2-Depth	DepthForge	67.03	48.27*	71.10
	Ours	**69.39**	**52.37**	**72.21**

Table 2: Comparison with SOTA. * denotes re-implementation with official code. **Bold** and underline denote the best and second-best. Depth denotes DepthAnythingV2. SYNTHIA has only 16 of the 19 evaluation classes in other datasets and we follow [71] to report the result over 19 classes, while some methods may report the result over 16 classes.

Method	Backbone	Trained on GTAV (G)				Trained on SYNTHIA (S)				Trained on Cityscapes (C)		
		→C	→B	→M	Avg.	→C	→B	→M	Avg.	→B	→M	Avg.
Classic DGSS												
SAN-SAW [48]	ResNet-101	45.33	41.18	40.77	42.43	40.87	35.98	37.26	38.04	54.73	61.27	58.00
AdvStyle [74]	ResNet-101	39.52	36.39	36.10	37.34	37.59	27.45	31.76	32.27	–	–	–
CMFormer [5]	Swin-L	55.31	49.91	60.09	55.10	44.59	33.44	43.25	40.43	59.27	71.10	65.19
Single foundation model												
Rein [60]	DINOv2	66.40	60.40	66.10	64.30	48.59	44.42	48.64	47.22	65.00	72.30	68.65
FADA [4]	DINOv2	68.23	61.94	68.09	66.09	50.04	45.83	49.86	48.58	65.12	75.86	70.49
SET [66]	DINOv2	68.06	61.64	67.68	65.79	49.65	45.45	49.45	48.18	65.07	75.67	70.37
SoMA [69]	DINOv2	71.82	61.31	71.67	68.27	-	-	-	-	**67.02**	76.45	71.74
tqdm [45]	EVA02-L	68.88	59.18	70.10	66.05	-	-	-	-	64.72	76.15	70.44
CLOUDS [2]	ConvNext-L	60.20	57.40	67.00	61.53	53.40	47.00	**55.80**	52.07	-	-	-
Multi-model depth-aligned fusion												
MFuser [71]	DINOv2-CLIP	71.24	61.08	71.14	67.82	53.16	45.59	52.27	50.34	65.58	78.10	71.84
MFuser [71]	DINOv2-SigLIP	71.10	61.19	71.71	68.00	54.14*	46.68*	53.57*	51.46*	65.44	77.97	71.71
MFuser* [71]	SAM-CLIP	63.49	58.27	68.10	63.29	50.08	41.73	48.97	46.93	60.26	73.16	66.71
Vireo [11]	DINOv2-Depth	70.69	**62.91**	69.63	67.74	-	-	-	-	66.73	75.99	71.36
DepthForge [12]	DINOv2-Depth	69.04	62.82	69.22	67.03	51.06*	45.87*	47.89*	48.27*	66.20	76.00	71.10
DepthForge [12]	EVA02-Depth	68.00	61.70	67.50	65.73	-	-	-	-	62.00	73.00	67.50
Multi-model cross-depth fusion (ours)												
Ours	DINOv2-CLIP	72.37	62.31	**73.52**	**69.40**	**55.09**	**48.64**	55.23	**52.99**	66.03	**78.98**	**72.51**
Ours	DINOv2-SigLIP	72.79	62.03	73.07	69.30	54.51	47.69	54.51	52.24	65.43	78.84	72.14
Ours	SAM-CLIP	64.78	58.29	68.33	63.80	50.65	41.13	48.05	46.61	60.18	72.28	66.23
Ours	DINOv2-Depth	**73.20**	61.98	73.00	69.39	54.17	48.25	54.69	52.37	65.74	78.68	72.21

4.2 Comparison with State-of-the-art DGSS methods

We compare SALC with various DGSS methods: (1) classic methods, including SAN-SAW [48], AdvStyle [74], and CMFormer [5]; (2) Single foundation model methods, including tqdm [45], Rein [60], SET [66], FADA [4], CLOUDS [2] and SoMA [69]; and (3) Multi foundation model methods, including MFuser [71], DepthForge [12] and Vireo [11].

We report results on **G→{C, B, M}**, **S→{C, B, M}**, and **C→{B, M}** in Table 2. SALC achieves competitive performance across all three settings, outperforming both single-foundation-model and depth-aligned multi-model baselines on most backbone configurations. On G→{C, B, M}, SALC achieves an average of 69.40 with DINOv2-CLIP, surpassing the previous depth-aligned baseline MFuser by 1.58 mIoU and the strongest single-foundation-model method (SoMA) by 1.13 mIoU. The improvement is larger on the challenging S→{C, B, M}, where SALC reaches 52.99 with DINOv2-CLIP, surpassing MFuser by +2.65. Notably, while multi-model baselines like DepthForge and MFuser sometimes underperform single-encoder CLOUDS on this setting, indicating that naive depth-aligned fusion may hurt generalisation under large domain shifts. SALC's effectiveness holds across multiple backbone combinations, reaching the largest single-domain gain of +4.16 mIoU over DepthForge on G→C. We observe that SALC's improvements on SAM-CLIP are smaller and occasionally slightly negative. We provide further analysis in Appendix E. We also provide qualitative comparisons in Figure 8.

4.3 In-depth layer selection analysis

In this section, we analyse the layer selection behaviour of SALC across domains and backbones. Our analysis reveals three key findings: the learned selection (1) leverages diverse candidate layers

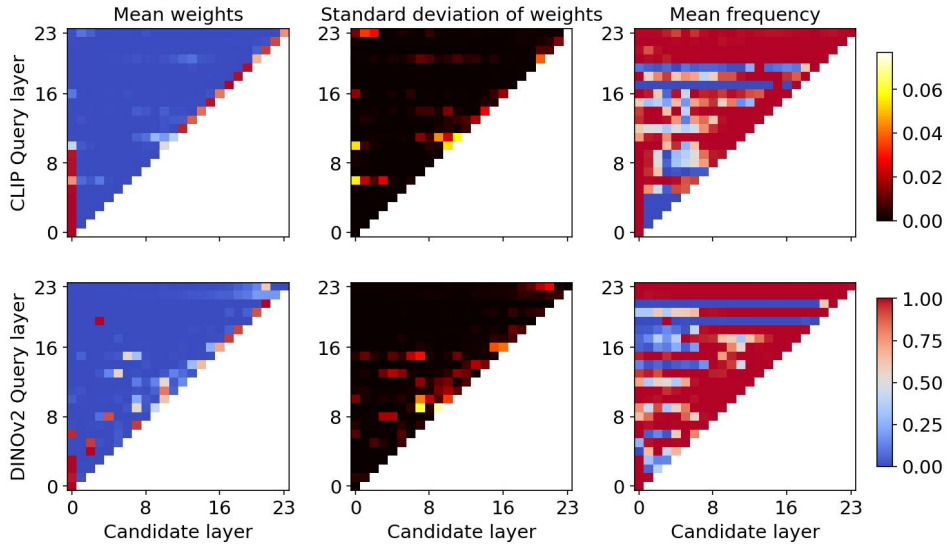


Figure 3: Layer selection statistics of SALC across domains (G, C, B, M).

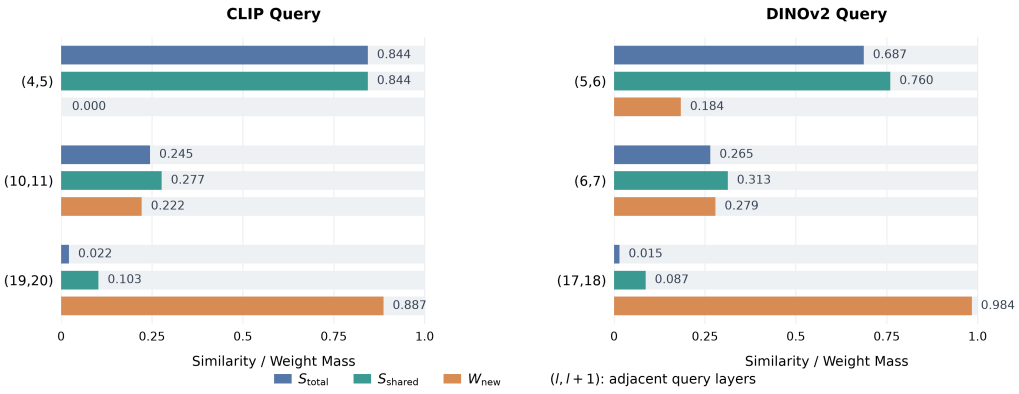


Figure 4: Representative patterns of weight redistribution across adjacent query layers.

without collapsing onto a fixed subset, (2) exhibits backbone-specific preferences that align with the complementary roles of VFM and VLM, and (3) remains largely consistent across domains. We further provide visual evidence on how candidate dropout reshapes the selection and discuss how it may benefit generalisation. We use DINOv2-CLIP combination on the G→{C, B, M} setting.

What patterns emerge in SALC's layer selection? Here, we analyse SALC's layer selection behaviour from three complementary perspectives.

(i) Layer-wise selection patterns. To gain a global perspective of selection patterns, we visualise the mean selection weights using $\mathbf{w}_t$, their standard deviations, and selection frequencies across different layers in Figure 3. From the mean weight, we observe SALC develops specific patterns for different backbones. CLIP primarily leverages DINOv2's shallow layers together with the depth-aligned one, while DINOv2 prefers to draw from CLIP's middle layers. This asymmetry reflects the complementary roles of the two backbones, where DINOv2's shallow layers extract local structure that CLIP lacks, while CLIP's middle layers exhibit higher transferability [55]. The standard deviations further show that the selection is stable across domains, with cross-domain disagreement concentrated only on a small subset of layers. Moreover, the frequency reveals that most candidates are occasionally selected, indicating that the token-wise sparsity control explores the full candidate pool rather than collapsing onto a fixed subset.

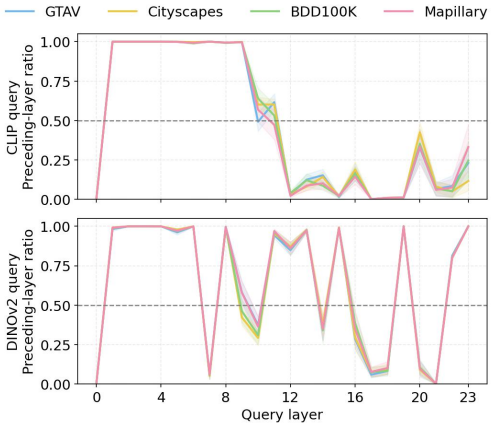


Figure 5: Preceding weight ratio across depth.

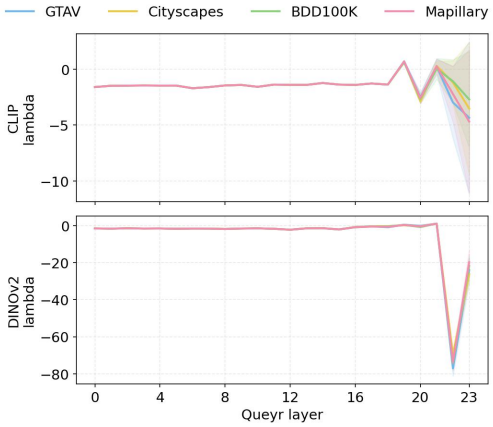


Figure 6: λ value across layers.

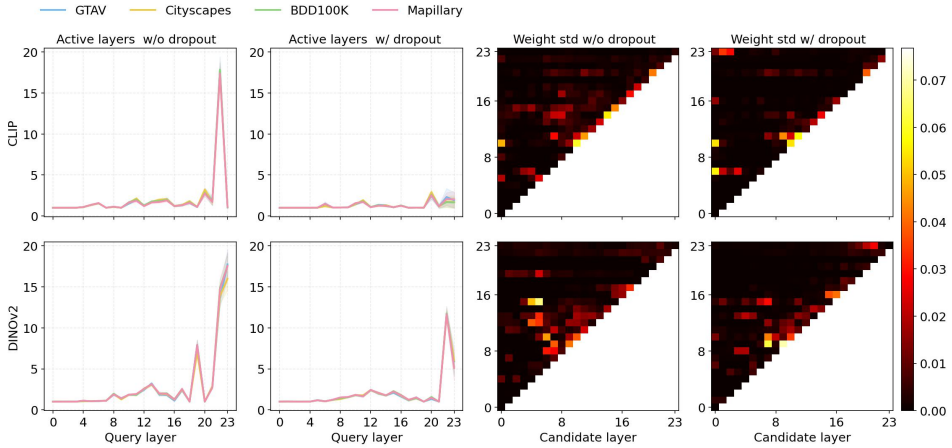


Figure 7: Active layers and cross-domain standard deviation before and after dropout.

(ii) Weight redistribution across layers. We further investigate how selection weights evolve as new candidate layers become available. Specifically, we consider three metrics: (1) the weight distribution similarity across all candidates (S_{total}), (2) distribution similarity across shared candidates (S_{shared}), and (3) the weight mass assigned to the new candidate (W_{new}). As shown in Figure 4, SALC exhibits three representative patterns: (1) adjacent query layers retain similar selection preferences (first row), (2) selection weights are redistributed among existing candidates (second row), and (3) the deeper query layer prioritises the newly available candidate (third row).

(iii) Preference for previous vs. current layers. We also report the ratio of preceding-layer weight to total weight in Figure 5. Most query layers show a non-zero preceding ratio, confirming that SALC consistently incorporates information from preceding layers. The ratio varies substantially across depths while remaining largely consistent across domains, indicating that SALC adapts its selection at each depth rather than learning a fixed rule. Moreover, some mid and deeper query layers exhibit notable cross-domain shifts. For mid layers, the shift may stem from their inherently stronger transferability [55, 43], which allows SALC to adjust their selection more freely depending on the target domain. For deeper layers, this is likely because they have a larger candidate pool and form the final representation for decoding, which provides them more flexibility.

How does SALC adapt sparsity across layers and domains? Figure 6 shows the depth-wise distribution of the learned sparsity factor λ across the four domains. We observe that λ exhibits substantially higher cross-domain variability at deeper layers while remaining stable at shallow ones. We believe this is because deeper query layers tend to encode richer high-level information, which may require finer per-token sparsity adjustment to handle semantics that vary across classes and

Table 3: Effect of masking selected layers.

Method	C	B	M	Avg.	Δ
MFuser	71.24	61.08	71.14	67.82	−1.58
Ours	72.37	62.31	73.52	69.40	–
Mask *most*	70.79	59.44	71.11	67.11	−2.29
Mask *least*	**72.46**	**62.37**	**73.52**	**69.45**	+0.05

Table 4: Extension to asymmetric architecture.

Method	C	B	M	Avg.	Δ
MFuser sym.	71.24	61.08	71.14	67.82	−1.58
Ours sym.	**72.37**	**62.31**	**73.52**	**69.40**	–
MFuser asym.	67.02	57.67	69.01	64.56	−0.64
Ours asym.	**67.04**	**58.76**	**69.81**	**65.20**	–

Table 5: Ablation on sparsity control and dropout. SALC: token-wise λ + frequency dropout.

Variant	→C	→B	→M	Avg.
Sparsity control				
Fixed λ	70.62	60.63	72.27	67.84
Layer-wise λ	70.67	60.25	72.61	67.84
Dropout				
No dropout	72.18	61.71	73.05	68.98
Uniform	71.18	60.13	72.24	67.85
SALC	**72.37**	**62.31**	**73.52**	**69.40**

Table 6: Ablation on candidate scope and fusion direction. SALC: all candidates + bidirectional.

Variant	→C	→B	→M	Avg.
Candidate scope				
Window size = 2	71.56	62.09	73.44	69.03
Window size = 4	71.65	**62.83**	73.37	69.28
Fusion direction				
CLIP → DINOv2 only	69.54	60.08	71.34	66.99
DINOv2 → CLIP only	71.71	60.68	73.07	68.49
SALC	**72.37**	62.31	**73.52**	**69.40**

spatial locations. This pattern is also consistent with the cross-domain shift observed in the preceding ratio at deeper layers, indicating that domain variation in SALC is more visible at deeper layers through both weight assignment and sparsity control.

How does dropout regularisation affect layer selection? Figure 7 compares the average active-layer count and cross-domain weight std before and after applying our dropout regularisation. After regularisation, the number of active candidates is clearly reduced and the selection becomes more concentrated. The cross-domain std of the selection is also reduced, as shown in the fourth column. Beyond compression, dropout exhibits distinct behaviour across query layers and models. At layers that are already moderately sparse, it does not further reduce the active-layer number but instead reorganises the dominant candidates, redirecting the selection toward different informative layers.

Based on these observations, we suggest two complementary reasons why dropout improves generalisation. First, by strengthening sparsity and lowering cross-domain variation, dropout guides SALC to commit to the most consistently informative candidates rather than spreading weight across many candidates that may carry domain-specific noise. Second, by perturbing dominant candidates during training, dropout encourages SALC to discover diverse informative layers, leading to a more robust selection that does not depend on any specific layer.

Does layer selection identify critical layers? We first use 500 images randomly sampled from the source domain to find a top-3 most-selected set of $\text{DINOv2}_{\text{most}} = \{21, 0, 20\}$ and $\text{CLIP}_{\text{most}} = \{21, 20, 7\}$, and a top-3 least-selected set of $\text{DINOv2}_{\text{least}} = \{2, 1, 4\}$ and $\text{CLIP}_{\text{least}} = \{17, 3, 16\}$. We then mask the *most* and *least* at inference, respectively. As shown in Table 3, masking the *most* set causes an obvious drop of 2.29 mIoU, falling below the depth-aligned baseline, while masking the *least* leaves performance essentially unchanged. This provides evidence that the selection discriminates informative from uninformative candidates rather than distributing weight arbitrarily.

Can SALC be extended to models with different depths? To improve the scalability, we can use a map function to handle two encoders $\mathcal{E}_A$ and $\mathcal{E}_B$ with different depths L_A and L_B. Given a query layer at depth l of $\mathcal{E}_A$, we map it to layer $\lfloor l \cdot \frac{L_B}{L_A} \rfloor$ of $\mathcal{E}_B$, a query layer then accesses all candidates up to its mapped depth, and vice versa. As shown in Table 4, we conduct experiments using a 24-layer CLIP-Large pairing with a 12-layer DINOv2-Base (asym.) and with a 24-layer DINOv2-Large (sym.). The results show that SALC outperforms the strongest baseline by 0.64 mIoU and 1.58 mIoU.

4.4 Ablation studies

We conduct ablation studies to examine each component in SALC by using the DINOv2-CLIP configuration on the G→{C, B, M} setting. More ablations are shown in Appendix F.

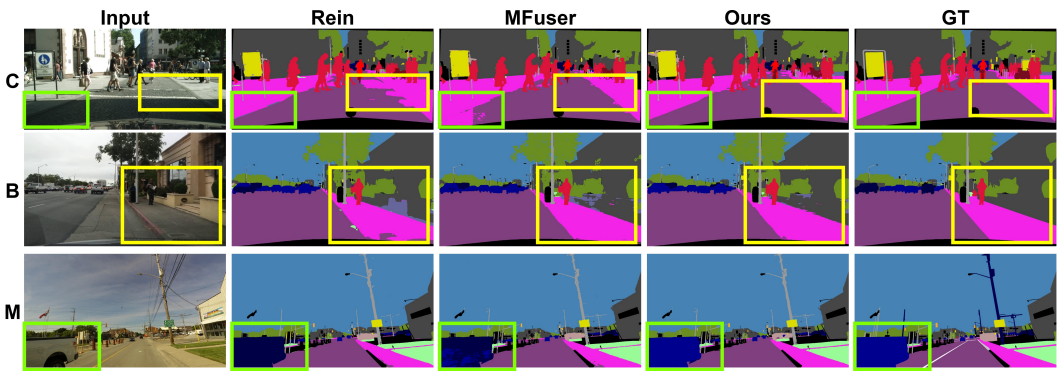


Figure 8: Qualitative comparison with Rein [60] and MFuser [71] under the G→{C, B, M} setting.

Table 7: Effect of fixed layer selection.

Method	→C	→B	→M	Avg.
MFuser	71.24	61.08	71.14	67.82
Fixed Top-2	70.32	60.76	71.68	67.59
Fixed Top-4	70.44	60.67	71.08	67.40
SALC	**72.37**	**62.31**	**73.52**	**69.40**

Table 8: Comparison of fusion strategies.

Variant	→C	→B	→M	Avg.
Depth-aligned	70.17	60.96	71.38	67.50
Cross att.	71.85	60.25	72.37	68.16
Cross Top2 routing	71.66	59.91	71.73	67.77
SALC	**72.37**	**62.31**	**73.52**	**69.40**

Sparsity control. We first study the sparsity factor λ in SALC. Specifically, we compare a fixed λ, a dynamic layer-wise λ and our dynamic token-wise λ. The fixed λ is set to the averaged λ learned by SALC for a fair comparison. As shown in Table 5, the other two variants reduce average mIoU by 1.56. Fixed λ removes input-dependent sparsity control, ignoring the growth of candidate layers. Layer-wise λ loses adaptivity at the token level, which is suboptimal for dense prediction tasks.

Candidate dropout regularisation. We further evaluate the proposed candidate dropout regularisation by comparing uniform and frequency-based dropout. As shown in Table 5, uniform dropout can harm generalisability because it drops candidates indiscriminately, unlike frequency-based dropout, which targets over-used candidates to strengthen sparsity and explore alternative informative layers. We further provide multiple runs of both dropout strategies in Appendix C, where frequency-based dropout shows smaller variance across runs.

Dynamic selection. We also investigate whether dynamic layer selection remains necessary when the most frequently selected layers are already known. We identify the top-2 and top-4 most-selected candidates per query layer on the source domain and retrain DINOv2–CLIP with each query layer restricted to the corresponding subset. As shown in Table 7, the two variants drop by 1.81 and 2.00 mIoU, respectively. While top-selected candidates carry generally robust features, restricting aggregation to them can discard necessary information retained in the remaining layers.

Candidate scope and fusion direction. Beyond the core SALC mechanism, we further study the candidate scope and the cross-model fusion direction. For the candidate scope, we restrict SALC to use only the most recent 2 or 4 layers as candidate layers. For fusion direction, we disable fusion from CLIP or DINOv2 by removing SALC in their respective encoders. As reported in Table 6, reducing the candidate scope hurts performance, dropping the average mIoU by 0.37 and 0.12 for window sizes of 2 and 4, respectively. This suggests that broader access to multi-level features and bidirectional fusion benefit generalisation.

Cross-layer fusion. To evaluate whether our sparse all-layer fusion is more effective than existing designs, we compare four fusion strategies: (1) depth-aligned fusion (linear connector), (2) cross-depth fusion (attention), (3) cross-depth fusion (Top2 routing), and (4) cross-depth fusion (SALC). As shown in Table 8, cross-depth fusion outperforms depth-aligned fusion by leveraging multi-level features, but generalises less well than SALC on this benchmark due to the lack of dynamic sparsity.

5 Conclusion

In this paper, we propose a Sparse All-Layer Connector for multi-model fusion in domain generalised semantic segmentation. Unlike existing methods that fuse features only at the same depth, SALC enables each layer to leverage features from all preceding layers of the other model. To avoid redundancy that may harm generalisation, we adopt Sparsegen with a token-wise, content-dependent sparsity factor for adaptive selection across layers and tokens. In addition, we introduce a frequency-based dropout regularisation that strengthens sparsity, lowers cross-domain variation, and encourages exploration of diverse layers. We evaluate SALC on five foundation models spanning three model types, achieving competitive performance across three evaluation settings and demonstrating the effectiveness of cross-layer fusion. Beyond performance gains, our analysis further reveals that SALC learns non-trivial, backbone-specific selection patterns that adapt across depths while remaining largely consistent across domains.

References

[1] Woo-Jin Ahn, Geun-Yeong Yang, Hyun-Duck Choi, and Myo-Taeg Lim. Style blind domain generalized semantic segmentation via covariance alignment and semantic consistence contrastive learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3616–3626, 2024.

[2] Yasser Benigmim, Subhankar Roy, Slim Essid, Vicky Kalogeiton, and Stéphane Lathuilière. Collaborating foundation models for domain generalized semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3108–3119, 2024.

[3] Qi Bi, Jingjun Yi, Huimin Huang, Hao Zheng, Haolan Zhan, Yawen Huang, Yuexiang Li, Xian Wu, and Yefeng Zheng. Nightadapter: Learning a frequency adapter for generalizable night-time scene segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23838–23849, 2025.

[4] Qi Bi, Jingjun Yi, Hao Zheng, Haolan Zhan, Yawen Huang, Wei Ji, Yuexiang Li, and Yefeng Zheng. Learning frequency-adapted vision foundation model for domain generalized semantic segmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 94047–94072, 2024.

[5] Qi Bi, Shaodi You, and Theo Gevers. Learning content-enhanced mask transformer for domain generalized urban-scene segmentation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 819–827, 2024.

[6] Yue Cao, Yangzhou Liu, Zhe Chen, Guangchen Shi, Wenhai Wang, Danhuai Zhao, and Tong Lu. MMFuser: Multimodal multi-layer feature fuser for fine-grained vision-language understanding. *arXiv preprint arXiv:2410.11829*, 2024.

[7] Prithvijit Chattopadhyay, Kartik Sarangmath, Vivek Vijaykumar, and Judy Hoffman. Pasta: Proportional amplitude spectrum training augmentation for syn-to-real domain generalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19288–19300, 2023.

[8] Gongwei Chen, Leyang Shen, Rui Shao, Xiang Deng, and Liqiang Nie. Lion: Empowering multimodal large language model with dual-level visual knowledge. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 26530–26540, 2024.

[9] I-Hsiang Chen, Hua-En Chang, Wei-Ting Chen, Jenq-Neng Hwang, and Sy-Yen Kuo. Exploring probabilistic modeling beyond domain generalization for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21755–21765, 2025.

[10] Kaibing Chen, Dong Shen, Hanwen Zhong, Huasong Zhong, Kui Xia, Di Xu, Wei Yuan, Yifei Hu, Bin Wen, Tianke Zhang, et al. EVLM: An efficient vision-language model for visual understanding. *arXiv preprint arXiv:2407.14177*, 2024.

[11] Siyu Chen, Ting Han, Chengzheng Fu, Changshe Zhang, Chaolei Wang, Jinhe Su, Guorong Cai, and Meiliu Wu. Leveraging depth and language for open-vocabulary domain-generalized semantic segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[12] Siyu Chen, Ting Han, Changshe Zhang, Xin Luo, Meiliu Wu, Guorong Cai, and Jinhe Su. Stronger, steadier & superior: Geometric consistency in depth vfm forges domain generalized semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8285–8295, 2025.

[13] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, 2022.

[14] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11575–11585, 2021.

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[16] Jian Ding, Nan Xue, Gui-Song Xia, Bernt Schiele, and Dengxin Dai. Hgformer: Hierarchical grouping transformer for domain generalized semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15413–15423, 2023.

[17] Mohammad Fahes, Tuan-Hung Vu, Andrei Bursuc, Patrick Pérez, and Raoul De Charette. A simple recipe for language-guided domain generalized segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 23428–23437, 2024.

[18] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 372–391, 2022.

[19] Jiaxing Huang, Dayan Guan, Aoran Xiao, and Shijian Lu. Fsdr: Frequency space domain randomization for domain generalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6891–6902, 2021.

[20] Wei Huang, Chang Chen, Yong Li, Jiacheng Li, Cheng Li, Fenglong Song, Youliang Yan, and Zhiwei Xiong. Style projected clustering for domain generalized semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3071, 2023.

[21] Christoph Hümmer, Manuel Schwonberg, Liangwei Zhou, Hu Cao, Alois Knoll, and Hanno Gottschalk. Vltseg: Simple transfer of clip-based vision-language representations for domain generalized semantic segmentation. *arXiv preprint arXiv:2312.02021*, 2023.

[22] Seogkyu Jeon, Kibeom Hong, and Hyeran Byun. Exploiting domain properties in language-driven domain generalization for semantic segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20791–20801, 2025.

[23] Yuru Jia, Lukas Hoyer, Shengyu Huang, Tianfu Wang, Luc Van Gool, Konrad Schindler, and Anton Obukhov. Dginstyle: Domain-generalizable semantic segmentation with image diffusion models and stylized semantic control. In *European Conference on Computer Vision (ECCV)*, pages 91–109, 2024.

[24] Dongsheng Jiang, Yuchen Liu, Songlin Liu, Jin'e Zhao, Hao Zhang, Zhen Gao, Xiaopeng Zhang, Jin Li, and Hongkai Xiong. From CLIP to DINO: Visual encoders shout in multi-modal large language models. *arXiv preprint arXiv:2310.08825*, 2023.

[25] Xueying Jiang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Domain generalization via balancing training difficulty and model capability. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18947–18957, 2023.

[26] Oğuzhan Fatih Kar, Alessio Tonioni, Petra Poklukar, Achin Kulshrestha, Amir Zamir, and Federico Tombari. Brave: Broadening the visual encoding of vision-language models. In *European Conference on Computer Vision (ECCV)*, pages 113–132, 2024.

[27] Jin Kim, Jiyoung Lee, Jungin Park, Dongbo Min, and Kwanghoon Sohn. Pin the memory: Learning to generalize semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4340–4350, 2022.

[28] Sunghwan Kim, Dae-hwan Kim, and Hoseong Kim. Texture learning domain randomization for domain generalized segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 677–687, 2023.

[29] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3992–4003, 2023.

[30] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, pages 3519–3529, 2019.

[31] Anirban Laha, Saneem A Chemmengath, Priyanka Agrawal, Mitesh M Khapra, Karthik Sankaranarayanan, and Harish G Ramaswamy. On controllable sparse alternatives to softmax. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6423–6433, 2018.

[32] Suhyeon Lee, Hongje Seong, Seongwon Lee, and Euntai Kim. Wildnet: Learning domain generalized semantic segmentation from the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9926–9936, 2022.

[33] Xiaotong Li, Yongxing Dai, Yixiao Ge, Jun Liu, Ying Shan, and LINGYU DUAN. Uncertainty modeling for out-of-distribution generalization. In *International Conference on Learning Representations (ICLR)*, 2022.

[34] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 3530–3543, 2026.

[35] Junyan Lin, Haoran Chen, Yue Fan, Yingqi Fan, Xin Jin, Hui Su, Jinlan Fu, and Xiaoyu Shen. Multi-layer visual feature fusion in multimodal llms: Methods, analysis, and best practices. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4156–4166, 2025.

[36] Xin Kun Lin, Ruoyu Guo, Jiaqi Guo, Maurice Pagnucco, and Yang Song. Hierarchical prompt injector for domain generalization segmentation. In *European Conference on Computer Vision (ECCV)*, pages 297–315, 2026.

[37] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, Jiaming Han, Siyuan Huang, Yichi Zhang, Xuming He, Hongsheng Li, and Yu Qiao. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

[38] Fengyuan Lu, Zixuan Duan, Xunzhi Xiang, Zhicheng Zhang, Wenbin Li, Yang Gao, and Qi Fan. Qprompt-r1: Real-time reasoning for domain-generalized semantic segmentation via group-relative query alignment. In *International Conference on Learning Representations (ICLR)*, 2026.

[39] Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. In *International Conference on Learning Representations (ICLR)*, volume 2025, 2025.

[40] Lingchen Meng, Jianwei Yang, Rui Tian, Xiyang Dai, Zuxuan Wu, Jianfeng Gao, and Yu-Gang Jiang. Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 23464–23487, 2024.

[41] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4990–4999, 2017.

[42] Hongwei Niu, Linhuang Xie, Jianghang Lin, and Shengchuan Zhang. Exploring semantic consistency and style diversity for domain generalized semantic segmentation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 6245–6253, 2025.

[43] Ambroise Odonnat, Vasilii Feofanov, Laetitia Chapel, Romain Tavenard, and Ievgen Redko. Layer by layer, module by module: Choose both for optimal ood probing of vit. *arXiv preprint arXiv:2603.05280*, 2026.

[44] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud 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. *https://arxiv.org/abs/2304.07193*, 2023.

[45] Byeonghyun Pak, Byeongju Woo, Sunghwan Kim, Dae-hwan Kim, and Hoseong Kim. Textual query-driven mask transformer for domain generalized segmentation. In *European Conference on Computer Vision (ECCV)*, pages 37–54, 2024.

[46] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In *European Conference on Computer Vision (ECCV)*, pages 484–500, 2018.

[47] Xingang Pan, Xiaohang Zhan, Jianping Shi, Xiaoou Tang, and Ping Luo. Switchable whitening for deep representation learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1863–1871, 2019.

[48] Duo Peng, Yinjie Lei, Munawar Hayat, Yulan Guo, and Wen Li. Semantic-aware domain generalized segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2584–2595, 2022.

[49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763, 2021.

[50] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems (NeurIPS)*, pages 12116–12128, 2021.

[51] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*, pages 102–118, 2016.

[52] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, 2016.

[53] Leyang Shen, Gongwei Chen, Rui Shao, Weili Guan, and Liqiang Nie. Mome: Mixture of multimodal experts for generalist multimodal large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 42048–42070, 2024.

[54] Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, Yilin Zhao, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, Bryan Catanzaro, Andrew Tao, Jan Kautz, Zhiding Yu, and Guilin Liu. Eagle: Exploring the design space for multimodal llms with mixture of encoders. In *International Conference on Learning Representations (ICLR)*, 2025.

[55] Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *International Conference on Machine Learning (ICML)*, pages 55854–55875, 2025.

[56] Qi Sun, Marc Pickett, Aakash Kumar Nain, and Llion Jones. Transformer layers as painters. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 25219–25227, 2025.

[57] PeiYuan Tang, Xiaodong Zhang, Chunze Yang, Haoran Yuan, Jun Sun, Danfeng Shan, and Zijiang James Yang. Unleashing the power of visual foundation models for generalizable semantic segmentation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 20823–20831, 2025.

[58] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 87310–87356, 2024.

[59] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9568–9578, 2024.

[60] Zhixiang Wei, Lin Chen, Yi Jin, Xiaoxiao Ma, Tianle Liu, Pengyang Ling, Ben Wang, Huaian Chen, and Jinjin Zheng. Stronger fewer & superior: Harnessing vision foundation models for domain generalized semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28619–28630, 2024.

[61] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Lili Ju, and Song Wang. Siamdoge: Domain generalizable semantic segmentation using siamese network. In *European Conference on Computer Vision (ECCV)*, pages 603–620, 2022.

[62] Qi Xu, Liang Yao, Zhengkai Jiang, Guannan Jiang, Wenqing Chu, Wenhui Han, Wei Zhang, Chengjie Wang, and Ying Tai. Dirl: Domain-invariant representation learning for generalizable semantic segmentation. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 2884–2892, 2022.

[63] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10371–10381, 2024.

[64] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems (NeurIPS)*, pages 21875–21911, 2024.

[65] Huanjin Yao, Wenhao Wu, Taojiannan Yang, Yuxin Song, Mengxi Zhang, Haocheng Feng, Yifan Sun, Zhiheng Li, Wanli Ouyang, and Jingdong Wang. Dense connector for mllms. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 33108–33140, 2024.

[66] Jingjun Yi, Qi Bi, Hao Zheng, Haolan Zhan, Wei Ji, Yawen Huang, Yuexiang Li, and Yefeng Zheng. Learning spectral-decomposited tokens for domain generalized semantic segmentation. In *ACM International Conference on Multimedia (ACM MM)*, pages 8159–8168, 2024.

[67] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2633–2642, 2020.

[68] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2100–2110, 2019.

[69] Seokju Yun, Seunghye Chae, Dongheon Lee, and Youngmin Ro. Soma: Singular value decomposed minor components adaptation for domain generalizable representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25602–25612, 2025.

[70] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11941–11952, 2023.

[71] Xin Zhang and Robby T Tan. Mamba as a bridge: Where vision foundation models meet vision language models for domain-generalized semantic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14527–14537, 2025.

[72] Dong Zhao, Jinlong Li, Shuang Wang, Mengyao Wu, Qi Zang, Nicu Sebe, and Zhun Zhong. Fishertune: Fisher-guided robust tuning of vision foundation models for domain generalized segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15043–15054, 2025.

[73] Yuyang Zhao, Zhun Zhong, Na Zhao, Nicu Sebe, and Gim Hee Lee. Style-hallucinated dual consistency learning for domain generalized semantic segmentation. In *European Conference on Computer Vision (ECCV)*, pages 535–552, 2022.

[74] Zhun Zhong, Yuyang Zhao, Gim Hee Lee, and Nicu Sebe. Adversarial style augmentation for domain generalized urban-scene segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 338–350, 2022.

A Active layer selection

Given importance scores $\mathbf{z}_t \in \mathbb{R}^l$ over the l candidate layers, Sparsegen determines the active layers by solving the following closed-form transformation:

$$\mathbf{w}_t = \underset{\mathbf{w}_t \in \mathbb{R}^l}{\arg\min} \|\mathbf{w}_t - \mathbf{z}_t\|_2^2 - \lambda_t \|\mathbf{w}_t\|_2^2, \quad \text{s.t. } \mathbf{w}_t \geq 0,\ \mathbf{1}^\top \mathbf{w}_t = 1,\ \lambda_t < 1. \tag{6}$$

Proposition 1 (Closed-form Sparsegen, Proposition 0.1 in [31]) *Let $\mathbf{z}_t \in \mathbb{R}^l$ be the importance scores, and let $z_{t,(1)} \geq z_{t,(2)} \geq \cdots \geq z_{t,(l)}$ denote the sorted coordinates of $\mathbf{z}_t$. Then, for any sparsity factor $\lambda_t \in (-\infty, 1)$, the Sparsegen solution is given by:*

$$w_{t,i} = \left[\frac{z_{t,i} - \tau_t}{1 - \lambda_t}\right]_+, \quad \forall i \in [l], \tag{7}$$

where $[x]_+ = \max(x, 0)$, and the threshold τ_t is determined as

$$\tau_t = \frac{\sum_{i=1}^{j} z_{t,(i)} - 1 + \lambda_t}{j}, \quad j = \max\left\{ j \in [l] \;\middle|\; 1 - \lambda_t + j\, z_{t,(j)} > \sum_{i=1}^{j} z_{t,(i)} \right\}, \tag{8}$$

such that $\mathbf{w}_t$ lies on the probability simplex, i.e., $\sum_{i=1}^{l} w_{t,i} = 1$.

From the above closed-form solution, the active layer set $\mathcal{S}_t$ corresponds to the top-j largest importance scores, with $|\mathcal{S}_t| = j$.

B Implementation details

We provide the implementation details in Table 10. In general, we directly use the official source code to train each framework. Moreover, we follow prior work [71, 11, 12] to use DINOv2-L, CLIP-L, SigLIP-L, and DepthAnythingV2-L with 24 transformer layers. We also resize their patch embeddings to a 16×16 patch size to handle 512×512 inputs. We train our model on a single NVIDIA RTX 6000 Ada GPU with 48GB of memory. We also provide parameter analysis in Table 9.

C Stability of dropout

Since dropout introduces stochasticity during training, we further evaluate the stability of *uniform* and *frequency-based* dropout by reporting the mean and standard deviation of three independent runs in Table 11. Across both DINOv2-CLIP and DINOv2-SigLIP, frequency-based dropout achieves the highest average mIoU with the smallest standard deviation (≤ 0.11), while uniform dropout exhibits substantially larger variance (0.89 on DINOv2-CLIP and 0.30 on DINOv2-SigLIP). The instability of uniform dropout aligns with its design: it drops candidates indiscriminately, providing no consistent regularisation target across runs. In contrast, frequency-based dropout focuses regularisation on over-used candidates, yielding both stronger and more stable improvements.

D Visualisation of layer selection

We provide additional visualisations of the layer selection on DINOv2-SigLIP under the G→ {C, B, M} setting. The selection patterns are consistent with those of DINOv2-CLIP shown in the main paper, confirming that SALC's behaviour generalises across backbone combinations.

Table 9: Comparison of parameter counts across different methods.

Method	Ours	Vireo [11]	DepthForge [12]	MFuser [71]	Rein [60]	FADA [4]	SoMA [69]
Trainable params (M)	9.59	3.78	2.99	141.79	2.99	11.65	4.9

Table 10: Implementation details of SALC integrated into the MFuser [71] and DepthForge [12] frameworks. We follow the original training settings of each framework unless otherwise specified. SALC-specific hyperparameters are shared across both configurations.

Hyperparameter	MFuser-based	DepthForge-based
Encoder configurations		
$\mathcal{E}_A$	DINOv2/SAM	DINOv2
$\mathcal{E}_B$	CLIP / SigLIP	DepthAnythingV2
Decoder	Mask2Former	Mask2Former
Training		
Optimizer	AdamW	AdamW
Learning rate	1e-4	1e-4
Weight decay	1e-4	0.05
LR schedule	Poly	OneCycleLR
Warmup iterations	1,500	4,000
Total iterations	20,000	40,000
Batch size	2	5
Input resolution	512×512	512×512
SALC-specific (shared)		
EMA momentum	0.99	
Penalty sharpness γ	2	
Maximum drop rate $p_{\max}$	0.4	

Table 11: Ablation study of dropout strategies under different backbone combinations on the G→{C, B, M} setting.

Variant	DINOv2-CLIP				DINOv2-SigLIP			
	→C	→B	→M	Avg.	→C	→B	→M	Avg.
No dropout	71.81±0.52	61.47±0.34	73.41±0.50	68.90±0.12	72.46±0.04	61.85±0.23	72.55±0.15	68.95±0.14
Uniform dropout	72.01±0.76	61.23±0.95	73.20±1.05	68.81±0.89	71.89±0.55	61.53±0.06	**73.00±0.37**	68.81±0.30
Frequency-based dropout (Ours)	**72.03±0.36**	**62.39±0.40**	**73.44±0.08**	**69.29±0.11**	**72.46±0.35**	**62.27±0.24**	72.95±0.16	**69.23±0.07**

E Limitations

E.1 Limited improvement on SAM-CLIP

We hypothesise that the smaller gain of SALC on SAM-CLIP stems from the weak layer-wise structure of SAM features when paired with CLIP. As shown in Figure 13, we compare the layer-wise Centered Kernel Alignment (CKA) [30] between CLIP vs DINOv2 and CLIP vs SAM. CLIP vs DINOv2 exhibits a clear layer-dependent structure: shallow CLIP layers align strongly with shallow-to-mid DINOv2 layers, while deeper CLIP layers form a distinct, less similar region. This indicates that each DINOv2 layer carries distinguishable information, providing useful signals for SALC's selection. In contrast, CLIP vs SAM shows uniformly low similarity across nearly all layer pairs, with no comparable layer-wise structure. Such weak and uniform alignment makes it difficult for SALC to identify informative SAM layers for cross-model fusion. We attribute this to SAM being trained with a different objective (segmentation supervision rather than self-distillation as in DINOv2), which leads to features less compatible with CLIP's representation space.

E.2 Limited gain on real-to-real benchmark

Compared to the G→{C, B, M} and S→{C, B, M} settings, SALC's improvement on C→ {B, M} is smaller, with most of the gain concentrated on Mapillary. We attribute this to the smaller domain gap between real-world source and target domains, where low-level features from the source domain transfer effectively to the targets. SALC selects layers at the semantic level rather than preserving low-level features, which limits its advantage in this setting.

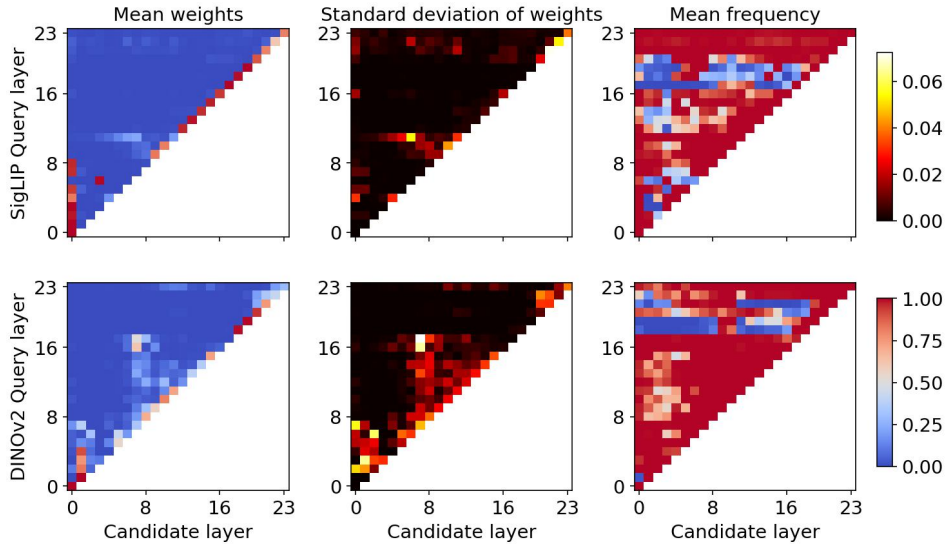


Figure 9: Layer selection statistics of SALC across domains (G, C, B, M) for DINOv2-SigLIP.

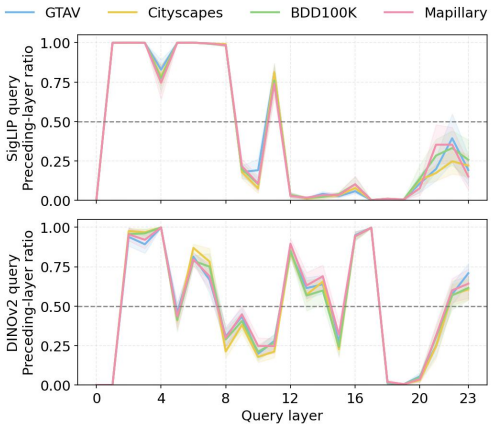


Figure 10: Preceding weight ratio across depth for DINOv2-SigLIP.

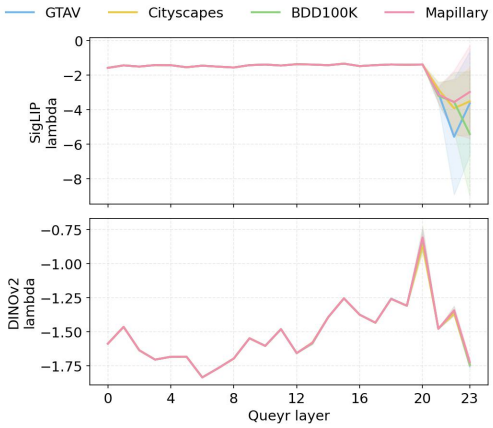


Figure 11: λ value across layers for DINOv2-SigLIP.

F Additional ablations

We provide additional ablations of SALC. Specifically, we study the choice of γ, the effect of EMA momentum, and the effect of applying RMSNorm before scoring. All experiments use the DINOv2–CLIP combination and are evaluated on the G→{C, B, M} setting.

The EMA momentum is used to track the running statistics for our *frequency-based dropout*, mitigating the noise introduced by the small batch size of 2. RMSNorm is applied to reduce the magnitude variations across depths in $\mathbf{F}_t$ before computing the importance scores $\mathbf{z}_t$. The hyperparameter γ controls the sharpness of the drop-rate penalty: with $\gamma > 1$, the penalty grows more sharply for over-used layers.

As reported in Table 12, EMA improves overall generalisation by stabilising the dropout regularisation, while RMSNorm enhances the effectiveness of importance scoring across candidate layers. For γ, we observe that a mildly larger penalty on frequently selected layers benefits generalisation, whereas an excessively large penalty becomes detrimental.

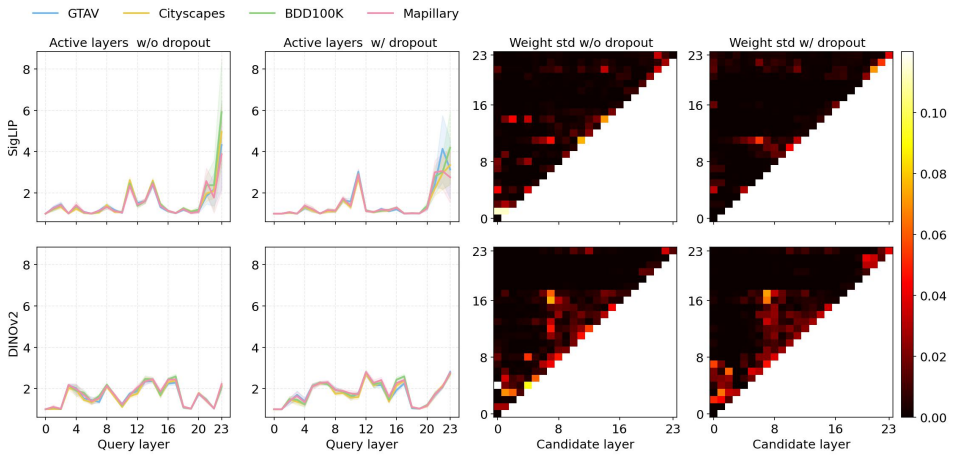


Figure 12: Active layers and cross-domain standard deviation before and after dropout for DINOv2-SigLIP.

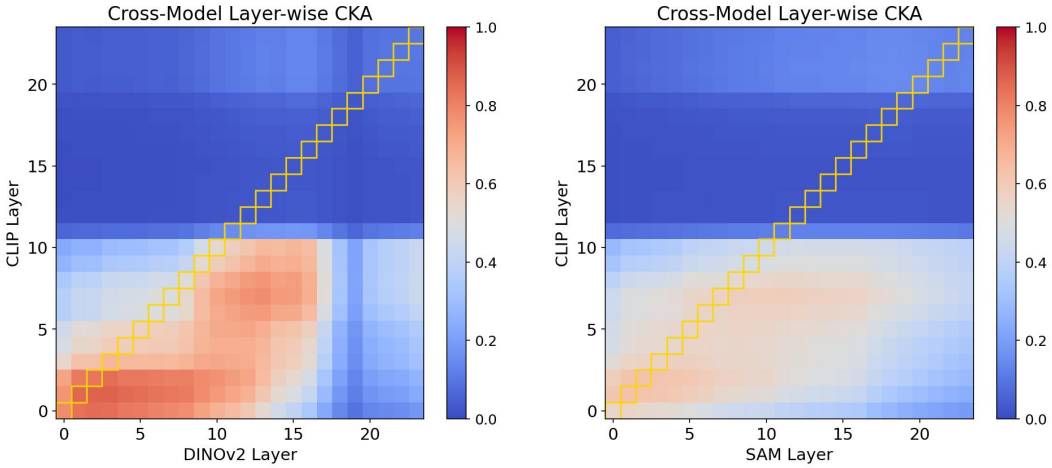


Figure 13: Cross-model layer-wise CKA between CLIP and different vision foundation models.

G Broader impacts

Our work studies domain generalisation in semantic segmentation, with potential applications in autonomous driving, robotics, and urban scene understanding. By improving model robustness across diverse, unseen environments, our method may help systems perform more reliably under real-world conditions such as adverse weather, nighttime, or unfamiliar scenes. However, segmentation models may still produce unreliable predictions on rare or out-of-distribution scenes. Deploying such systems requires careful evaluation, robust failure handling, and complementary safety mechanisms. Our work does not introduce additional risks beyond those already associated with semantic segmentation in general, and all experiments use publicly available academic datasets.

Table 12: Ablation studies on EMA, RMSNorm, and γ using DINOv2-CLIP under the G→{C, B, M} setting. SALC uses EMA, RMSNorm and $\gamma = 2$.

Variant	→C	→B	→M	Avg.
EMA				
w/o EMA	72.18	61.71	73.05	68.98
RMSNorm				
w/o RMSNorm	72.05	61.81	73.38	69.08
Penalty sharpness γ				
$\gamma = 1$	71.69	61.97	73.52	69.06
$\gamma = 3$	72.16	61.62	73.50	69.09
SALC	**72.37**	**62.31**	**73.52**	**69.40**

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We study the multi-model fusion in domain generalised semantic segmentation.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss our limitations in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This paper does not include theoretical contributions.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our method is built upon the official public code of prior work. We have released our code. All training details can be found in the open public code of prior work.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have released our code.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: We provide training details in the main paper and appendix. We use the default setting of prior work.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: For dropout regularisation, we report results over multiple runs. In addition, our motivation experiments are conducted with 40 random seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide training details in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss our broader impacts in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not involve the release of data or models with a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators of the code, datasets, and pretrained models used in this work are properly credited through citations. We use publicly available assets in accordance with their respective licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have released the code upon acceptance and provide appropriate documentation alongside it.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing experiments or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve research with human subjects, so IRB approval and participant risk disclosure are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is only used for editing and formatting.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.