S2E: DENSE FUNCTIONAL DIFFERENTIATION IN FEED-FORWARD NETWORKS VIA PROTOTYPE-GUIDED SUBSPACE MODULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) architectures increase model capacity through specialized experts and input-dependent routing, but incur parameter duplication, routing complexity, and irregular execution. This raises a fundamental question: can functional differentiation emerge within a single dense feed-forward network (FFN) without independent experts, token dispatch, or sparse activation? We answer this question affirmatively by proposing *Semantic Subspace Expert (S2E)*, a dense FFN that realizes functional differentiation by organizing a shared hidden representation into semantic subspaces and applying prototype-guided subspace modulation. Subspace weights combine global cosine similarity for directional subspace-level alignment with periodic coordinate-level correspondence for fine-grained feature matching. Across ImageNet-1K classification, distribution-shift benchmarks, and COCO detection and segmentation, S2E consistently improves accuracy–efficiency trade-offs across diverse vision backbones, reducing FLOPs in all evaluated configurations and reducing parameter counts in most cases. On DeiT-T, the compact S2E configuration achieves 74.9% Top-1 accuracy with 3.9M parameters and 0.90G FLOPs, improving the FFN-$4\times$ baseline by 2.7 percentage points. Controlled ablations and functional analyses further show that shared projection promotes representational diversity and that the learned subspaces exhibit non-redundant, category-dependent response preferences while remaining fully active during inference.

1 INTRODUCTION

Transformers have become the dominant architecture for visual representation learning due to their scalability, strong representation capability, and broad applicability across vision tasks (Vaswani et al., 2017; Dosovitskiy et al., 2020). Within Transformer blocks, feed-forward networks (FFNs) account for a substantial proportion of model parameters and computation, serving as fundamental components for feature transformation and representation learning. Recent studies have shown that FFNs are not merely homogeneous transformations; rather, their neurons and hidden dimensions exhibit structured activation patterns, input-dependent behaviors, and specialized functional responses (Geva et al., 2021; Zhang et al., 2022). In language models, FFN layers have further been interpreted as implicit key-value memories that encode semantic associations (Geva et al., 2021). These observations suggest that the internal organization of FFN representations plays an important role in how neural networks allocate and utilize representational capacity, motivating the study of differentiated functional behavior within dense representations.

A major paradigm for achieving functional specialization is Mixture-of-Experts (MoE), which increases model capacity through multiple expert transformations with input-dependent computation. Switch Transformers demonstrate that conditional expert activation enables efficient scaling to large models (Fedus et al., 2022), while vision MoE variants improve accuracy–capacity trade-offs through sparse or soft expert activation (Riquelme et al., 2021; Zhou et al., 2022; Puigcerver et al., 2024; Komatsuzaki et al., 2022). However, conventional MoE architectures achieve specialization through explicit parameter separation, where independently optimized experts are selected or weighted according to input characteristics. This design introduces parameter duplication, routing complexity, irregular memory access, and load-balancing challenges (Liu et al., 2024; Videau et al.,

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.

2024). Although router-free approaches reduce explicit routing overhead, they still maintain separate expert structures or independently organized representations (Roller et al., 2021; Lv et al., 2025; Jain et al., 2024). These limitations motivate us to investigate whether differentiated functional behavior can instead emerge within a shared dense representation, without introducing independent experts or sparse routing.

An alternative direction improves dense FFNs by reorganizing internal representations rather than introducing explicit experts. Grouped FFN methods divide hidden channels into predefined groups to improve efficiency (Li et al., 2022; Zhang et al., 2023), while dynamic gating methods adapt feature responses according to input characteristics (Gao et al., 2019; Liu et al., 2023). Other approaches derive expert-like structures through factorization, transformation, or dense-to-expert conversion after training (Zhang et al., 2022; Oldfield et al., 2024; Huang et al., 2025; Devvrit et al., 2024; Yang et al., 2024). Although these methods improve efficiency or adaptability, their specialization mechanisms rely on fixed partitioning, externally introduced structures, or post-hoc transformation. They do not directly investigate whether differentiated functional behavior can emerge through end-to-end learning within a shared dense representation.

Motivated by this possibility, we introduce *dense functional differentiation*, in which differentiated response patterns emerge through learned organization and adaptive modulation of a shared dense representation. Unlike conventional expert architectures that obtain specialization through parameter separation and discrete routing, dense functional differentiation treats differentiated behavior as an emergent property of learned representation organization. Based on this principle, we propose Semantic Subspace Expert (S2E), a dense FFN architecture that organizes a shared representation into subspaces whose functional responses emerge through end-to-end training and are adaptively modulated through prototype-guided similarity. S2E therefore provides expert-like differentiated behavior without explicit expert separation or sparse computation.

Our main contributions are summarized as follows:

- We introduce the principle of *dense functional differentiation* and instantiate it through *Semantic Subspace Expert (S2E)*, demonstrating that a shared dense projection can develop measurable, non-redundant subspace behaviors through learned organization and prototype-guided modulation, without explicit expert separation or sparse routing.
- We design a prototype-guided subspace modulation mechanism that combines global cosine similarity for directional subspace-level alignment with periodic coordinate-level correspondence for finer-grained feature matching. These complementary signals regulate subspace updates at different granularities while preserving fully dense execution.
- We conduct extensive experiments across diverse vision backbones, robustness benchmarks, and dense prediction tasks. Ablation studies, representation-diversity analysis, subspace intervention analysis, and functional-selectivity evaluation show that S2E improves accuracy–efficiency trade-offs while inducing measurable, non-redundant, category-dependent functional differentiation within dense representations.

2 Related Work

Mixture-of-Experts Architectures. Mixture-of-Experts (MoE) architectures achieve functional specialization by introducing multiple expert transformations with input-dependent computation. Switch Transformers demonstrate efficient scaling through sparse expert activation (Fedus et al., 2022), while vision MoE variants explore sparse, soft, and expert-choice routing to improve accuracy–capacity trade-offs (Riquelme et al., 2021; Zhou et al., 2022; Puigcerver et al., 2024; Komatsuzaki et al., 2022). However, these approaches obtain specialization through explicit expert separation and token-dependent assignment, resulting in parameter duplication, routing overhead, irregular memory access, and optimization challenges (Liu et al., 2024; Videau et al., 2024). Router-free methods, including Hash Layers, AoE, and MoNE, reduce routing complexity but still maintain independent expert structures or separated representations (Roller et al., 2021; Lv et al., 2025; Jain et al., 2024). Fine-grained approaches such as PEER and Product Key Memory further increase specialization granularity, but rely on explicit expert selection or retrieval mechanisms (He, 2024; Lample et al., 2019).

Recent methods further exploit similarity or subspace structures for routing. ERMoE learns expert-specific eigenbases for similarity-based routing, ProMoE introduces prototype-guided expert assign-

Table 1: Structural comparison of representative expert and dense FFN architectures. Global Proj.: shared global projection; Dense: all units active; Struct.: structured representation organization; E2E: end-to-end training.

Method	Granularity	Global Proj.	Dense	Struct.	E2E	Primary Goal
Standard MoE (Fedus et al., 2022)	Full FFN	×	×	×	✓	Capacity scaling
Soft MoE (Puigcerver et al., 2024)	Full FFN	×	×	×	✓	Soft expert assignment
AoE (Lv et al., 2025)	Full FFN	×	×	×	✓	Router-free experts
PEER (He, 2024)	Single-neuron expert	Partial	×	×	✓	Fine-grained experts
Grouped FFN (Li et al., 2022)	Channel group	×	✓	✓	✓	Static grouping
MoEfication (Zhang et al., 2022)	Neuron cluster	✓	×	×	×	Post-hoc extraction
DeRS (Huang et al., 2025)	Upcycled expert block	×	×	✓	×	Dense-to-MoE conversion
μMoE (Oldfield et al., 2024)	Factored expert block	×	×	✓	×	Compact experts
MatFormer (Devvrit et al., 2024)	Nested FFN	Partial	✓	✓	✓	Elastic inference
S2E (ours)	Semantic subspace	✓	✓	✓	✓	Dense functional differentiation

ment, and STAR incorporates evolving subspace structures to improve routing decisions (Cheng et al., 2026; Wei et al., 2026; Park & Park, 2026). These methods use similarity or subspace information to select among independent experts. In contrast, S2E investigates whether expert-like functional roles can emerge inside a shared dense FFN. Rather than selecting separated experts, S2E learns adaptive semantic subspaces within a common representation and keeps all subspaces jointly active during inference.

Dense FFN Organization and Adaptive Modulation. Recent studies improve dense FFNs by reorganizing hidden representations or introducing adaptive modulation. Grouped FFN methods partition hidden channels into predefined groups (Li et al., 2022; Zhang et al., 2023), while dynamic gating methods adapt feature responses according to input characteristics (Gao et al., 2019; Liu et al., 2023). Other approaches derive expert-like structures through factorization, post-hoc extraction, or dense-to-expert conversion (Zhang et al., 2022; Oldfield et al., 2024; Huang et al., 2025; Yang et al., 2024; Devvrit et al., 2024; Huang et al., 2023). Although these methods improve efficiency or adaptability, their specialization relies on fixed partitioning, input-dependent feature weighting, or externally constructed expert structures. They do not investigate whether functional specialization can emerge through end-to-end learning inside a shared dense representation. S2E differs by organizing a shared projection into structured subspaces and learning their functional behavior through prototype-guided modulation, enabling adaptive functional differentiation without explicit expert separation.

Prototype-Based Representation Learning. Prototype-based methods use representative vectors to organize and guide feature representations. Prototypical Networks employ class-level prototypes for metric-based recognition (Snell et al., 2017), while modern Hopfield networks utilize stored patterns and associative retrieval mechanisms (Ramsauer et al., 2020). In these approaches, prototypes mainly serve as references for recognition or memory retrieval. S2E adopts prototypes from a different perspective: learnable prototypes act as internal directional references that regulate subspace transformations during FFN computation. Rather than representing classes or retrieving stored patterns, S2E prototypes guide adaptive subspace updates and encourage differentiated functional preferences within a shared dense representation.

We compare representative architectures according to five structural properties: shared global projection, dense activation, structured organization, end-to-end training, and primary objective. Table 1 summarizes these differences. Existing approaches obtain specialization through expert separation, sparse activation, predefined grouping, or post-hoc transformation. S2E provides an alternative paradigm in which specialization emerges through learned organization of a shared representation rather than through independent expert parameters.

3 METHOD

Rather than achieving functional differentiation through independent expert parameters or sparse token routing, S2E develops differentiated functional roles within a shared representation space. It consists of two core components: (1) semantic subspace organization, which organizes a globally projected token representation into K semantic subspaces of dimension r; and (2) prototype-guided subspace modulation, which applies token-dependent updates to these subspaces using learnable prototypes. All subspaces remain active in every forward pass; no discrete selection or token dispatch is performed. The entire module operates as a drop-in replacement for a standard Transformer FFN block.

Semantic Subspace Organization. Given an input token representation $\mathbf{x} \in \mathbb{R}^d$, S2E maps it to a shared hidden representation of dimension $H = Kr$ through a single global projection:

$$\mathbf{z} = \phi\left(\mathbf{x}W_{\text{in}}^{\top}\right), \qquad \mathbf{z} \in \mathbb{R}^H,$$

where ϕ denotes the GELU activation. The hidden representation is then reshaped into K semantic subspaces of dimension r:

$$\mathbf{x}_k = \text{reshape}(\mathbf{z})_k \in \mathbb{R}^r, \qquad k = 1, \ldots, K.$$

For each token, S2E computes a shared subspace centroid $\mathbf{c} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{x}_k \in \mathbb{R}^r$. The centroid is computed dynamically during the forward pass without exponential-moving-average accumulation or stop-gradient. It provides a compact summary of the token's overall subspace state and serves as the common reference for prototype-based similarity scoring. This organization creates structured functional units within a shared representation space while avoiding separate projection spaces for individual subspaces.

Prototype-Guided Subspace Modulation. Each layer maintains a bank of K learnable prototype vectors $\{\mathbf{s}_k \in \mathbb{R}^r\}_{k=1}^{K}$, with one prototype associated with each semantic subspace. For each token, we compute a similarity score between the token-wise subspace centroid $\mathbf{c}$ and each prototype. The score combines two complementary signals:

$$A_k = a_k^{(1)} \cdot \left(1 + \kappa \cdot a_k^{(2)}\right), \quad \gamma_k = \text{softmax}_k(A_k/\tau),$$

where $a_k^{(1)}$ measures global directional similarity and $a_k^{(2)}$ captures finer coordinate-level correspondence. The global similarity is computed using cosine similarity:

$$a_k^{(1)} = \frac{\mathbf{s}_k^{\top}\mathbf{c}}{\|\mathbf{s}_k\|_2\|\mathbf{c}\|_2 + \epsilon}, \quad a_k^{(2)} = \frac{1}{r}\sum_{j=1}^{r} \cos\left(\rho\left(s_{k,j} - c_j\right) + \theta_{k,j}\right),$$

where ρ controls the periodic scale and $\theta_{k,j}$ are learnable phase offsets per subspace per coordinate. The resulting similarity scores are converted into modulation weights through softmax with temperature τ. Each semantic subspace is then updated toward its corresponding prototype:

$$\mathbf{x}'_k = \phi\left(\mathbf{x}_k + \gamma_k\left(\mathbf{s}_k - \mathbf{x}_k\right)\right),$$

Finally, all subspaces are concatenated and projected to the output dimension:

$$\mathbf{y} = \text{flatten}(\{\mathbf{x}'_k\}_{k=1}^{K})W_{\text{out}} + b_{\text{out}}.$$

All subspaces are updated in every forward pass, preserving fully dense computation. Prototypes and phase offsets are learned end-to-end via backpropagation alongside the projection matrices.

Design Principles and Intuitions. The S2E design is motivated by three principles that distinguish it from both MoE and grouped FFN approaches.

First, *shared projection with structured subspace organization* balances parameter efficiency and functional differentiation. Unlike MoE, which uses separate transformations for different experts, S2E uses a single global projection that avoids parameter duplication and places all subspaces in a common representational space. Functional differentiation therefore emerges through adaptive modulation rather than separate expert parameters.

Second, *prototype-guided adaptive modulation* provides input-dependent functional differentiation without discrete selection. Unlike MoE routing, which assigns tokens to selected expert modules, S2E regulates the update strength of every subspace toward its corresponding prototype. Thus, all subspaces remain active for every token, while their relative modulation weights vary according to the input. This enables differentiated functional roles while preserving fully dense computation.

Third, the *cosine-plus-periodic similarity mechanism* captures complementary forms of correspondence. Cosine similarity measures the global directional relationship between the shared centroid and each prototype, whereas the periodic term captures finer coordinate-wise correspondence through learnable phase offsets. These two signals therefore operate at different levels of granularity: the cosine component characterizes overall directional alignment, while the periodic component provides coordinate-level modulation based on prototype–centroid differences. Their combination provides complementary information for determining subspace modulation strength, consistent with the improvements observed in the weighting-mechanism ablation.

4 EXPERIMENTS

We design experiments to evaluate S2E from three complementary perspectives: (1) whether S2E improves the accuracy–efficiency trade-off across diverse architectures and tasks; (2) whether each proposed component contributes to the observed improvements and whether shared projection provides advantages over independent subspace projections; and (3) whether the learned subspaces exhibit measurable functional differentiation beyond simple representation diversity.

4.1 EXPERIMENTAL SETUP

Datasets and Protocols. We train all classification models on ImageNet-1K for 300 epochs, following the DeiT training recipe (Touvron et al., 2021) with AdamW, weight decay 0.05, a cosine learning-rate schedule, and a 20-epoch warm-up. Data augmentation includes RandAugment, Mixup, CutMix, and random erasing. We evaluate robustness on four distribution-shift benchmarks: ImageNet-C (Hendrycks & Dietterich, 2019) (mean corruption error; lower is better), ImageNet-A (Hendrycks et al., 2021b), ImageNet-R (Hendrycks et al., 2021a), and ImageNet-Sketch (Recht et al., 2019) (classification accuracy; higher is better). Object detection and instance segmentation use COCO 2017 (Lin et al., 2014) with Mask R-CNN FPN under a $1\times$ schedule (12 epochs). Detailed training configurations, augmentation settings, implementation details, and latency measurement protocols are provided in Appendix A.

Backbones. S2E replaces only the Transformer FFN block while keeping all other components unchanged. We evaluate S2E across representative vision backbones covering diverse architecture families and scales, including DeiT-T/S/B (Touvron et al., 2021), ViT-S (Dosovitskiy et al., 2020), Swin-T/S/B(Liu et al., 2021), PoolFormer-S12 (Yu et al., 2022), PVTv2-B1 (Wang et al., 2022), and LV-ViT-S/M (Yang et al., 2022). All baseline and S2E-enhanced models use identical training protocols to isolate the effect of FFN reorganization.

4.2 MAIN PERFORMANCE RESULTS

Baseline Comparison on DeiT-T. We first compare S2E against dense, grouped, and MoE-style FFN variants on DeiT-T under matched training budgets (Table 2). With $K = 24$ subspaces of width $r = 16$ ($H = Kr = 384$), S2E reaches 74.9% Top-1 accuracy using 3.9M parameters and 0.90G FLOPs. Relative to the standard FFN-$4\times$ baseline, this yields a $+2.7$-point improvement with 31.6% fewer parameters and 28.6% fewer FLOPs. Grouped FFN reaches 72.8% at 5.2M parameters and 1.05G FLOPs, whereas Soft MoE and Router-free MoE achieve 71.8% and 72.7%, respectively, while requiring substantially more parameters and computation. Post-hoc MoEfication reaches 68.9% accuracy, indicating that the gains of S2E are not explained by post-hoc expert extraction alone. The full comparison, including latency, throughput, and peak memory, is reported in Table 16 (Appendix C). Under the same hardware protocol, S2E maintains a favorable accuracy–efficiency trade-off while avoiding separate expert execution.

Table 2: Performance comparison on DeiT-T ImageNet-1K.

Method	Params (M)	FLOPs (G)	Top-1 (%)
Standard FFN (Touvron et al., 2021)	5.7	1.26	72.2
Grouped FFN (Li et al., 2022)	5.2	1.05	72.8
Soft MoE (Puigcerver et al., 2024)	23.4	1.66	71.8
Router-free MoE (Liu et al., 2026)	21.8	1.80	72.7
Post-hoc MoEfication (Zhang et al., 2022)	5.7	0.94	68.9
S2E (Ours)	3.9	0.90	74.9

Cross-Backbone Generalization. Table 3 summarizes the effect of replacing the standard FFN with S2E under the configurations used for each cross-backbone evaluation. Across the representative backbones reported in Table 3, S2E improves ImageNet-1K accuracy by $+0.6$ to $+3.6$ percentage points while reducing FLOPs in every case and reducing parameter count for all models except DeiT-T, where the parameter count remains unchanged. The largest clean-accuracy gains are observed for DeiT-T ($+3.6$), PoolFormer-S12 ($+3.4$), and DeiT-S ($+2.1$), while Swin-T and

Swin-S improve by $+2.0$ and $+0.6$ points, respectively. Under distribution shift, S2E also improves robustness across all five backbones, as reflected by lower ImageNet-C mCE. In particular, on DeiT-T ($K = 48$, $r = 16$), S2E reduces mCE from 71.1 to 66.2, increases ImageNet-R accuracy from 32.6% to 38.5%, and improves ImageNet-Sketch accuracy from 20.2% to 23.1%. Complete results, including ImageNet-A, latency, and throughput, are provided in Table 17 (Appendix D.1).

Table 3: Performance and efficiency changes after replacing the standard FFN with S2E on representative vision backbones. Changes are computed relative to the corresponding standard backbone.

Backbone	ΔParams	ΔFLOPs	ΔIN-1K	ΔmCE ↓
DeiT-T	0.0%	−11.1%	+3.6	−4.9
DeiT-S	−24.4%	−25.6%	+2.1	−1.3
Swin-T	−38.9%	−13.3%	+2.0	−5.6
Swin-S	−35.1%	−29.1%	+0.6	−2.8
PoolFormer-S12	−38.7%	−33.3%	+3.4	−4.8

Transfer to dense prediction. To evaluate whether the benefits of S2E extend beyond image classification, we integrate S2E into Swin-T and Swin-S backbones and evaluate object detection and instance segmentation on COCO 2017 using Mask R-CNN with FPN. Under identical training protocols, replacing the standard FFN with S2E improves both detection and segmentation performance while reducing model size and computational cost. For Swin-T, box AP increases from 43.7 to 46.3 and mask AP from 39.8 to 40.5. For Swin-S, box AP increases from 49.9 to 51.7 and mask AP from 41.6 to 42.9. These improvements are accompanied by reductions in parameters and FLOPs for both backbones, indicating that the benefits of S2E extend beyond image classification to spatially structured prediction tasks while maintaining favorable accuracy–efficiency trade-offs. Full detection and segmentation results, including throughput measurements, are provided in Table 18 (Appendix D.2).

4.3 Component Ablation Studies

We ablate the weighting mechanism and projection organization on DeiT-T to examine the contribution of the principal S2E design choices. The ablations are evaluated under the same model configuration and training protocol used for the main experiments.

Weighting-mechanism ablation. Table 4 evaluates whether adaptive similarity-based modulation is necessary for effective subspace differentiation. Uniform weighting reaches 72.4%, whereas cosine-only and periodic-only modulation improve accuracy to 73.2% and 72.8%, respectively. Combining the two signals yields the best performance of 74.9%, a $+2.5$-point gain over uniform weighting. These results support the core design motivation of S2E: global directional similarity and coordinate-level periodic correspondence provide complementary signals for input-dependent modulation, enabling more effective organization of the shared subspace representation. The ablation establishes the contribution of the weighting mechanism, while functional differentiation is examined separately through the diversity, intervention, and selectivity analyses.

Table 4: Weighting-mechanism ablation on DeiT-T.

Weighting variant	Top-1 (%)	Rel. delay
Uniform weighting ($\gamma_k = 1/K$)	72.4	3.2%
Input-side MLP weighting	72.9	9.1%
Cosine-only	73.2	5.1%
Periodic-similarity-only	72.8	4.7%
Cosine + entropy reg. (aux. loss)	73.4	6.2%
Cosine + periodic (ours)	**74.9**	**7.8%**

Shared versus independent projection. Table 5 compares the shared S2E projection with independent per-subspace projections at comparable model scale. The shared projection achieves higher accuracy (74.9% vs. 74.1%) and greater representation diversity ($\delta = 0.61$ vs. 0.42), while also

using slightly fewer parameters (3.9M vs. 4.0M) and lower computation (0.91G vs. 1.02G FLOPs). These results indicate that separating the subspace projections provides no advantage in either predictive performance or representational differentiation. Instead, the shared projection provides a common representational basis within which prototype-guided modulation can induce differentiated subspace responses. This supports the central design hypothesis that functional differentiation can emerge through structured organization of a shared dense representation rather than requiring independent expert parameterizations.

Table 5: Shared versus independent subspace projections on DeiT-T. δ denotes the mean pairwise cosine distance between subspace outputs.

Projection	Params (M)	FLOPs (G)	Top-1 (%)	Diversity δ
Independent per-subspace	4.0	1.02	74.1	0.42
Shared (S2E)	3.9	0.90	**74.9**	**0.61**

4.4 Representation and Functional Differentiation Analysis

Performance gains alone do not establish that the learned subspaces are functionally differentiated. We therefore examine three progressively stronger questions: (i) are subspaces *non-redundant*? (ii) does each subspace contribute *causally*? and (iii) do subspaces exhibit *category-dependent* preferences?

Representation diversity. We measure subspace differentiation using the average pairwise cosine distance δ between subspace outputs. On DeiT-T, S2E achieves $\delta = 0.61$, compared with 0.42 for the independent-projection variant and 0.28 for the standard FFN. The stage-aggregated analysis in Table 11(Appendix B.7) shows higher average diversity in later network stages, increasing from approximately $\delta = 0.35$ in the early-stage aggregate to $\delta = 0.67$ in the late-stage aggregate. This stage-level trend should not be interpreted as a monotonic layer-by-layer increase: the block-specific analysis in Figure 3 reveals substantial local variation across depth. Together, these analyses indicate that subspace organization evolves across the network, with the overall stage-level separation becoming stronger toward later stages while individual blocks may exhibit different degrees of alignment.

Subspace intervention analysis. To test whether individual subspaces make non-redundant contributions, we suppress each of the 48 subspaces independently during inference ($\mathbf{x}'_k = 0$) and measure the resulting change in ImageNet-1K accuracy. Every subspace produces a measurable accuracy drop, ranging from 0.30% to 1.20% (mean 0.72%), indicating that no individual subspace is completely redundant. The graded variation in intervention effects further suggests that different subspaces contribute unequally to the final prediction rather than behaving as duplicated representations. Complete intervention statistics and grouped-masking controls are provided in Table 12 and Table 15 (Appendix B.5.2- B.7), with corresponding activation and representation-similarity visualizations shown in Figs. 1–3.

Table 6: Functional-selectivity analysis of DeiT-T S2E on ImageNet-1K ($K = 48$, $r = 16$).

Metric	Result
High-selectivity subspaces ($SI > 1.5$)	47/48 (97.9%)
Maximum selectivity index SI	6.44
Average selectivity index SI	2.30
Accuracy drop on preferred classes	0.181%
Accuracy drop on non-preferred classes	0.056%
Preferred/non-preferred sensitivity ratio	3.24×

Functional selectivity. Finally, we examine whether individual subspaces exhibit category-dependent response preferences. For each subspace k in the selectivity-analysis configuration ($K = 48$, $r = 16$), we compute a Selectivity Index $SI_k = \max_c a_{k,c}/\overline{a_k}$ from its mean activation

magnitude across classes. We then identify highly selective subspaces, suppress each of them independently during inference, and compare the resulting accuracy decrease on preferred versus non-preferred class subsets. As summarized in Table 6, 47 of 48 subspaces (97.9%) satisfy $SI > 1.5$, with an average SI of 2.30 and a maximum of 6.44. Suppressing these subspaces produces a larger mean accuracy decrease on preferred classes (0.181%) than on non-preferred classes (0.056%), corresponding to a sensitivity ratio of approximately $3.23\times$. In a complementary analysis, the learned prototypes achieve a clustering purity of 0.625 across five coarse ImageNet class groups. Together with the representation-diversity and subspace-intervention results above, these findings provide evidence that the learned subspaces exhibit measurable category-dependent response preferences while remaining jointly active during inference.

4.5 Capacity and Sensitivity Analysis

At fixed total hidden dimension $H = Kr = 384$, varying the number of subspaces K reveals a non-monotonic optimum: too few subspaces limit differentiation, whereas too many reduce the representational capacity available to each subspace. The best trade-off is achieved at $K = 24$, $r = 16$ (Table 7, Appendix B.1). S2E is also robust to its free hyperparameters: across the soft-assignment temperature τ, periodic scale ρ, and alignment weight κ, Top-1 accuracy varies by less than 0.5 percentage points (Table 8, Appendix B.2). Among these, τ has the largest observed influence, whereas performance remains comparatively stable across ρ and κ, indicating that S2E does not depend on narrow hyperparameter tuning. Scaling the subspace dimension r further shows a gradual accuracy–efficiency trade-off as capacity decreases, and S2E at $H = 192$ retains accuracy comparable to the standard FFN-$4\times$ baseline while using 59.6% fewer parameters and 56.3% fewer FLOPs (Table 10, Appendix B.4). Finally, operator-level profiling shows that centroid computation and similarity scoring account for the largest share of S2E block runtime (42%), while prototype-directed subspace updates account for an additional 7%. These implementation-dependent costs may be further reduced through optimized kernels; the operator-wise breakdown is reported in Table 13 (Appendix B.6).

5 Discussion

5.1 Mechanistic Insights

Our experiments provide three insights into how dense functional differentiation emerges in S2E.

First, *shared projection can facilitate rather than constrain functional differentiation*. Compared with independent per-subspace projections, the shared projection achieves higher accuracy and greater representation diversity. This indicates that differentiated subspace behavior does not necessarily require isolated parameter spaces; instead, a shared representational basis can support distinct subspace responses through prototype-guided modulation. S2E therefore obtains expert-like differentiation through organization and modulation of shared representations rather than explicit parameter separation.

Second, *prototype-guided modulation provides an alternative between uniform dense transformation and sparse expert routing*. Conventional FFNs apply a common transformation pathway to all tokens, whereas MoE architectures specialize through selective activation of independent experts. S2E instead keeps all subspaces jointly active while adaptively modulating their transformations according to prototype similarity, enabling input-dependent functional differentiation without discrete routing or token dispatch.

Third, *the cosine-plus-periodic similarity mechanism provides complementary forms of correspondence*. Cosine similarity captures the global directional relationship between the shared centroid and each prototype, whereas the periodic term captures finer coordinate-wise correspondence through learnable phase offsets. Their combination provides complementary information for subspace modulation, consistent with the weighting-mechanism ablation.

5.2 Broader Interpretation

The results suggest that functional differentiation need not fundamentally depend on increasing the number of independent experts. S2E shows that a single dense representation can develop measurable, non-redundant subspace behaviors when its internal organization is structured and adaptively modulated. This offers a complementary perspective to MoE architectures, where specialization is obtained through parameter separation and expert selection.

Across the evaluated architectures, improvements in predictive performance, robustness, and computational efficiency are consistent with more effective utilization of the available representational capacity rather than gains arising simply from increased model size. In particular, S2E improves accuracy and robustness while reducing FLOPs, and for DeiT-T does so without increasing the parameter count. The magnitude of these gains varies across backbones, suggesting that the benefits of S2E depend partly on the underlying architecture and representational capacity.

Moreover, the depth-wise analyses reveal a structured but non-monotonic evolution of subspace organization. The stage-aggregated diversity measure in Table 11 shows greater average representational separation in later network stages, whereas the block-specific analysis in Figure 3 reveals substantial local variation, including increased alignment at intermediate depth followed by renewed separation in the final block. These complementary observations suggest that functional differentiation evolves with network depth rather than following a uniform layer-by-layer trajectory. Diversity alone, however, does not establish the semantic role of individual subspaces.

5.3 Limitations and open directions

First, the current understanding of functional differentiation remains mainly at the category level. Although representation-diversity analysis, subspace interventions, and functional-selectivity measurements reveal differentiated subspace behaviors, the precise semantic meaning of individual subspaces remains unclear. Future work could investigate finer-grained roles associated with visual primitives, attributes, and transformations.

Second, although S2E reduces FLOPs and achieves lower measured latency across the evaluated backbones, its dynamic similarity computation introduces additional implementation complexity. Profiling indicates that centroid and similarity computation constitute a substantial portion of the S2E runtime. Optimized kernels and hardware-aware implementations could therefore further improve deployment efficiency.

Third, current evaluations focus on vision architectures. Although the proposed principle is not restricted to vision, whether prototype-guided subspace organization transfers to language models and other modalities remains unverified. Extending S2E beyond vision is therefore a natural direction for future work.

Fourth, the learning dynamics underlying subspace differentiation remain insufficiently understood. Although experiments show that differentiated subspace behaviors emerge through end-to-end optimization, a theoretical analysis of how prototypes, subspaces, and modulation signals interact to produce differentiation remains an open question.

Finally, the relationship between dense functional differentiation and sparse computation remains unexplored. Future architectures could combine S2E-style shared representation organization with selective computation to investigate whether dense differentiation and sparse efficiency provide complementary benefits.

6 Conclusion

We presented S2E, a dense feed-forward network that enables functional differentiation through prototype-guided subspace modulation within a shared projection. Without independent expert parameters, token routing, or sparse execution, S2E improves the accuracy–efficiency trade-off across diverse vision backbones, distribution-shift benchmarks, and dense prediction tasks, while consistently reducing FLOPs and reducing parameter counts in most evaluated configurations. Controlled ablations identify semantic subspace organization and the complementary cosine-plus-periodic similarity mechanism as important components of the proposed design. Representation-diversity analysis, subspace intervention, and functional-selectivity evaluation further show that the learned subspaces exhibit non-redundant and category-dependent response patterns while remaining jointly active during inference. Together, these results provide empirical evidence that measurable functional differentiation can emerge through structured organization and adaptive modulation of a shared dense representation, without requiring independently parameterized experts or sparse token routing. Future work includes finer-grained functional analysis, evaluation beyond vision, kernel-level efficiency optimization, theoretical analysis of subspace differentiation, and hybrid designs that combine dense functional differentiation with selective computation.

7 AI USE DISCLOSURE

Generative AI tools were used to assist with the writing and editorial refinement of this manuscript, including improvements to grammar, readability, sentence structure, and academic presentation. Generative AI was not used to generate experimental data, formulate the proposed method, develop theoretical or mathematical claims, design or conduct experiments, or determine the reported results and conclusions. All AI-assisted content was reviewed, revised, and verified by the authors. The authors take full responsibility for the accuracy, originality, scientific claims, and final content of the manuscript.

REFERENCES

Uranik Berisha, Jens Mehnert, and Alexandru Paul Condurache. Efficient data driven mixture-of-expert extraction from trained networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20082–20091, 2025.

Anzhe Cheng, Shukai Duan, Shixuan Li, Chenzhong Yin, Mingxi Cheng, Heng Ping, Tamoghna Chattopadhyay, Sophia I Thomopoulos, Shahin Nazarian, Paul Thompson, et al. Ermoe: Eigen-reparameterized mixture-of-experts for stable routing and interpretable specialization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12997–13006, 2026.

Fnu Devvrit, Sneha Kudugunta, Sandeep Sinha, et al. Matformer: Nested transformer for elastic inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

William Fedus, Jeff Dean, et al. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Xitong Gao, Yuxin Zhao, Łukasz Dudziak, Robert Mullins, and Chengzhong Xu. Dynamic channel pruning: Feature boosting and suppression for efficient deep neural networks. In *International Conference on Learning Representations (ICLR)*, 2019.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5484–5495, 2021.

Xu Owen He. Mixture of a million experts. *arXiv preprint arXiv:2407.04153*, 2024.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8320–8329, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15262–15271, 2021b.

Yongqi Huang, Peng Ye, Xiaoshui Huang, Sheng Li, Tao Chen, Tong He, and Wanli Ouyang. Experts weights averaging: A new general training scheme for vision transformers. *arXiv preprint arXiv:2308.06093*, 2023.

Yongqi Huang, Peng Ye, Chenyu Huang, Jianjian Cao, Lin Zhang, Baopu Li, Gang Yu, and Tao Chen. Ders: Towards extremely efficient upcycled mixture-of-experts models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10056–10066. IEEE, 2025.

Nikhil P. Jain, Jiachen T. Wang, Xiang Li, Prateek Mittal, Sijia Liu, and Ravi Netravali. Mone: Mixture of nested experts for transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Aran Komatsuzaki, Joan Puigcerver, James Lee-Thorp, Carlos Riquelme Ruiz, Basil Mustafa, Joshua Ainslie, Yi Tay, Mostafa Dehghani, and Neil Houlsby. Sparse upcycling: Training mixture-of-experts from dense checkpoints. *arXiv preprint arXiv:2212.05055*, 2022.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Xiang Li, Wenhai Wang, Xiaolin Hu, and Yi Yang. G-ffn: Grouped feed-forward network for efficient vision transformers. *arXiv preprint arXiv:2206.02351*, 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 (ECCV)*, pp. 740–755. Springer, 2014.

Tianlin Liu, Mathieu Blondel, Carlos Riquelme, and Joan Puigcerver. Routers in vision mixture of experts: An empirical study. *arXiv preprint arXiv:2401.15969*, 2024.

Yilun Liu, Jinru Han, Sikuan Yan, Volker Tresp, and Yunpu Ma. Routing-free mixture-of-experts. *arXiv preprint arXiv:2604.00801*, 2026.

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 (ICCV)*, pp. 10012–10022, 2021.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. Dynamic gated feed-forward networks for vision transformers. *arXiv preprint arXiv:2303.07975*, 2023.

Ang Lv, Ruobing Xie, Yining Qian, Songhao Wu, Xingwu Sun, Zhanhui Kang, Di Wang, and Rui Yan. Autonomy-of-experts models. *arXiv preprint arXiv:2501.13074*, 2025.

James Oldfield, Markos Georgopoulos, Grigorios G. Chrysos, Christos Tzelepis, Yannis Panagakis, Mihalis A. Nicolaou, Jiankang Deng, and Ioannis Patras. Multilinear mixture of experts: Scalable expert specialization through factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 53022–53063, 2024.

Sumin Park and Noseong Park. Star: Rethinking moe routing as structure-aware subspace learning. *arXiv preprint arXiv:2606.08814*, 2026.

Joan Puigcerver, Carlos Riquelme Ruiz, Basil Mustafa, and Neil Houlsby. From sparse to soft mixtures of experts. In *International Conference on Learning Representations (ICLR)*, 2024.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Sandro Gröbl, Michael Feigl, et al. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2020.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Carlos Riquelme, Joan Puigcerver, Basil Mustafa, Maxim Neumann, Rodolphe Jenatton, André Susano Pinto, Daniel Keysers, and Neil Houlsby. Scaling vision with sparse mixture of experts. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 8583–8595, 2021.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 17555–17566, 2021.

Dai Shi. Transnext: Robust foveal visual perception for vision transformers. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17773–17783. IEEE, 2024.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13229–13239, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Mathurin Videau, Alessandro Leite, Marc Schoenauer, and Olivier Teytaud. Mixture of experts in image classification: What's the sweet spot? *arXiv preprint arXiv:2411.18322*, 2024.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022.

Yujie Wei, Shiwei Zhang, Hangjie Yuan, Yujin Han, Zhekai Chen, Jiayu Wang, Difan Zou, Xihui Liu, Yu Liu, Hongming Shan, et al. Routing matters in moe: Scaling diffusion transformers with explicit routing guidance. In *International Conference on Learning Representations*, volume 2026, pp. 23381–23405, 2026.

Junjie Wu, Qilong Wang, Jiangtao Xie, Pengfei Zhu, and Qinghua Hu. Asymmetric factorized bilinear operation for vision transformer. In *International Conference on Learning Representations (ICLR)*, pp. 77467–77487, 2025.

Xuwei Xu, Yang Li, Yudong Chen, Jiajun Liu, and Sen Wang. Repavit: Scalable vision transformer acceleration via structural reparameterization on feedforward network layers. *arXiv preprint arXiv:2505.21847*, 2025.

Yixing Xu, Chao Li, Dong Li, Xiao Sheng, Fan Jiang, Lu Tian, Ashish Sirasao, and Emad Barsoum. Enhancing vision transformer: Amplifying non-linearity in feed-forward network module. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235. PMLR, 2024.

Chenglin Yang, Yilin Wang, Jianming Zhang, He Zhang, Zijun Wei, Zhe Lin, and Alan Yuille. Lite vision transformer with enhanced self-attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11988–11998, 2022.

Yuqi Yang, Peng-Tao Jiang, Qibin Hou, Hao Zhang, Jinwei Chen, and Bo Li. Mole: Multi-task dense prediction via mixture of low-rank experts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27927–27937, 2024.

Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10809–10819, 2022.

Hao Zhang, Yufei Wang, Jian Liu, Qiang Zhang, and Xiaopeng Wei. Cg-ffn: Channel-guided grouped feed-forward network for lightweight visual recognition. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2023.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 877–890, 2022.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M. Dai, Quoc V. Le, James Laudon, et al. Mixture-of-experts with expert choice routing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 7103–7114, 2022.

A Experimental Details

Image classification training. We train all models on ImageNet-1K for 300 epochs following the DeiT training recipe (Touvron et al., 2021). Optimization uses AdamW with a base learning rate of 1×10^{-3}, weight decay of 0.05, a cosine learning-rate schedule, and a 20-epoch linear warm-up. Data augmentation includes RandAugment (magnitude 9), Mixup ($\alpha = 0.8$), CutMix ($\alpha = 1.0$), and random erasing ($p = 0.25$). Robustness is evaluated on ImageNet-C (Hendrycks & Dietterich, 2019), ImageNet-A (Hendrycks et al., 2021b), ImageNet-R (Hendrycks et al., 2021a), and ImageNet-Sketch (Recht et al., 2019). We report mean corruption error (mCE; lower is better) for ImageNet-C and classification accuracy (higher is better) for the remaining benchmarks.

Object detection and instance segmentation. ImageNet-pretrained backbones are transferred to COCO 2017 (Lin et al., 2014) using Mask R-CNN with an FPN backbone. Models are optimized with AdamW at an initial learning rate of 1×10^{-4} under the standard $1\times$ training schedule (12 epochs). We report bounding-box average precision (AP_b) and mask average precision (AP_m).

Latency protocol. Latency and throughput are measured on a single NVIDIA RTX 4090 GPU using PyTorch 2.8 and CUDA 12.8 with batch size 256 under FP16 precision. We perform 50 warm-up iterations followed by 200 timed iterations and report mean latency (standard deviation $< 1\%$) and throughput calculated as $256/\text{latency}$. These measurements are reported together with parameter counts and FLOPs because theoretical computation reduction does not always translate directly into wall-clock acceleration.

B Additional Experimental Results

B.1 Subspace granularity at fixed capacity

To investigate how the organization of semantic subspaces affects dense functional differentiation, we vary the number of subspaces K while keeping the total hidden dimension $H = Kr = 384$ fixed. Therefore, all variants maintain identical parameter counts and computational complexity, allowing performance differences to be attributed to the granularity of representation organization rather than additional model capacity. Table 7 shows a non-monotonic relationship between subspace granularity and recognition accuracy. When the number of subspaces is small ($K = 16$), each subspace has a larger dimensional capacity ($r = 24$), but the limited number of subspaces restricts the diversity of functional roles that can emerge. Increasing the granularity to $K = 24$ provides a better balance between the number of functional subspaces and the representation capacity of each subspace, achieving the best performance of 74.9% Top-1 accuracy.

Further increasing the number of subspaces ($K = 32, 48, 64$) gradually reduces the dimensional capacity of individual subspaces ($r = 12, 8, 6$), which limits their ability to capture sufficiently rich functional patterns and results in reduced performance. These observations indicate that dense functional differentiation does not simply benefit from increasing the number of subspaces; instead, effective specialization requires an appropriate balance between subspace diversity and per-subspace representational capacity. The results support the hypothesis that the organization of shared representations is a critical factor in enabling differentiated functional roles.

Table 7: Subspace granularity at fixed total hidden dimension $H = Kr = 384$ on DeiT-T.

K	r	$H = Kr$	Params (M)	FLOPs (G)	Top-1 (%)	Δ
16	24	384	3.9	0.90	72.8	—
24	16	384	3.9	0.90	**74.9**	**+2.1**
32	12	384	3.9	0.90	73.7	+0.9
48	8	384	3.9	0.90	72.9	+0.1
64	6	384	3.9	0.90	72.4	-0.4

B.2 HYPER-PARAMETER SENSITIVITY

Table 8 evaluates the sensitivity of the soft-assignment temperature τ, periodic scale ρ, and alignment weight κ. Accuracy remains within 0.5 points across the tested settings, indicating that S2E does not depend on narrow hyper-parameter tuning.

Table 8: Sensitivity to τ, ρ, and κ on DeiT-T (K=24, r=16). Defaults in bold.

τ	Top-1 (%)	ρ	Top-1 (%)	κ	Top-1 (%)
0.1	74.6	0.01	74.5	0.0	74.6
0.3	74.8	0.05	74.8	0.05	74.8
0.5	**74.9**	**0.1**	**74.9**	**0.1**	**74.9**
0.7	74.8	0.2	74.8	0.2	74.8
1.0	74.5	0.5	74.4	0.5	74.7

B.3 DEFAULT HYPER-PARAMETER SETTINGS

Table 9 summarizes the default hyper-parameter values used throughout all experiments. Unless otherwise specified, these settings are consistently applied across different backbone architectures and experimental comparisons to ensure fair evaluation. The selected values control the granularity of semantic subspaces, prototype-guided modulation strength, and assignment dynamics of S2E. These hyper-parameters are kept fixed in the main experiments, while their sensitivity is further analyzed through dedicated ablation studies.

Table 9: Default hyper-parameter settings.

Hyper-parameter	Default value
Number of subspaces K	24
Per-subspace dimension r	16
Soft-assignment temperature τ	0.5
Periodic scale ρ	0.1
Periodic alignment weight κ	0.1
Prototype initialization	Xavier uniform
Phase offset $\theta_{k,j}$ initialization	Zero

B.4 SUBSPACE DIMENSION SCALING

Fixing $K = 24$ while reducing the subspace dimension r progressively decreases the hidden representation size and computational cost. Table 10 shows that S2E maintains competitive accuracy across a range of subspace dimensions on DeiT-T. The full configuration ($r = 16$, $H = 384$) achieves the highest accuracy of 74.9%, while smaller r values provide progressively more efficient configurations with gradual accuracy degradation. Notably, even at $r = 8$ ($H = 192$), S2E reaches 72.3% Top-1 accuracy, outperforming the matched-width FFN-2$\times$ baseline (71.5%) and approximately matching the larger FFN-4$\times$ baseline (72.2%), while using substantially fewer parameters and FLOPs than the latter.

Table 10: Accuracy–efficiency trade-off via subspace dimension scaling on DeiT-T.

K	r	$H = Kr$	Expansion	Params (M)	FLOPs (G)	Top-1 (%)
Dense FFN-2$\times$ baseline			2.00$\times$	3.9	0.91	71.5
24	16	384	2.00$\times$	3.9	0.90	74.9
24	14	336	1.75$\times$	3.5	0.82	74.1
24	12	288	1.50$\times$	3.1	0.73	73.8
24	10	240	1.25$\times$	2.7	0.64	72.9
24	8	192	1.00$\times$	2.3	0.55	72.3

S2E exhibits a clear accuracy–efficiency trade-off as the subspace dimension is reduced. At $r = 16$, S2E achieves 74.9% Top-1 accuracy with 3.9M parameters and 0.90G FLOPs, while reducing r to 12 lowers the cost to 3.1M parameters and 0.73G FLOPs with 73.8% accuracy. At the most compressed setting, $r = 8$, the model uses only 2.3M parameters and 0.55G FLOPs, corresponding to 41.0% fewer parameters and 39.6% fewer FLOPs than the matched-width FFN-2$\times$ baseline, while retaining 72.3% accuracy. Relative to the larger FFN-4$\times$ baseline, this configuration achieves comparable accuracy with substantially lower computational cost. Overall, the gradual degradation across decreasing r suggests that S2E preserves effective representation differentiation under reduced capacity, with $r = 16$ providing the highest accuracy and smaller values offering progressively lower-cost alternatives.

B.5 Additional Functional-Differentiation Analysis

We provide additional analyses supporting the functional-differentiation results discussed in Section 4.4. Specifically, we report stage-aggregated representation diversity, individual-subspace intervention statistics, and grouped-masking controls to further examine whether the learned semantic subspaces exhibit non-redundant, differentiated, and category-dependent contributions. These complementary analyses characterize subspace organization at different levels, from broader stage-wise representation separation to the predictive effects of suppressing individual or grouped subspaces.

B.5.1 Stage-Aggregated Representation Diversity

Table 11 reports the average pairwise cosine distance δ between subspace outputs after aggregating layers into early, middle, and late network stages. The corresponding stage-level averages increase from 0.35 in the early stage to 0.52 in the middle stage and 0.67 in the late stage, indicating greater average representational separation in later stages.

These values summarize groups of layers and therefore do not imply a monotonic increase in diversity at every individual Transformer block. The block-specific visualization in Figure 3 instead reveals non-monotonic local variation in inter-subspace similarity. Taken together, the two analyses are complementary: the stage-aggregated results capture the broader depth-wise trend, while the block-level visualization shows that individual layers can exhibit different degrees of subspace alignment.

Table 11: Stage-aggregated representation diversity, measured as average pairwise cosine distance δ, in DeiT-T S2E. Higher δ indicates greater average separation between subspace representations.

Network stage	Average diversity δ
Early stage	0.35
Middle stage	0.52
Late stage	0.67
Overall	0.61

B.5.2 Complete causal-subspace intervention

To examine whether individual subspaces provide non-redundant contributions, we suppress each subspace independently during inference and measure the resulting ImageNet-1K accuracy change. Table 12 summarizes the complete intervention statistics. All 48 subspaces produce measurable accuracy changes, with drops ranging from 0.30% to 1.20% and an average drop of 0.72%. Moreover, the magnitude of the accuracy drop correlates with the selectivity index ($\rho = 0.78$), indicating that subspaces with stronger category-dependent responses tend to have greater influence on prediction.

B.6 Operator-Wise Runtime Breakdown

Table 13 reports the forward-pass time distribution of the principal components within an S2E block on a single RTX 4090 GPU. Centroid computation and similarity scoring account for the largest share of runtime (42%), while prototype-directed subspace updates account for an additional 7%. By comparison, reshape and subspace partitioning contribute only 4% of the measured block time.

Table 12: Subspace intervention analysis statistics on DeiT-T S2E; each subspace is suppressed individually.

Measurement	Result
Number of subspaces evaluated	48
Minimum accuracy drop (%)	0.30
Maximum accuracy drop (%)	1.20
Average accuracy drop (%)	0.72
Correlation with selectivity index (ρ)	0.78

These results indicate that the main implementation cost arises from similarity computation rather than from the structural partitioning of the shared hidden representation itself.

Table 13: Operator-wise runtime breakdown for an S2E block on DeiT-T ($K = 24$, $r = 16$).

Sub-module	Time fraction
Global input projection + GELU	31%
Reshape / subspace partitioning	4%
Centroid computation + similarity scoring	42%
Prototype-directed subspace update	7%
Concatenate + output projection	16%

B.7 Understanding functional differentiation

We further analyze functional differentiation through activation patterns, representation similarity, quantitative diversity measurements, and masking interventions.

Activation visualization. For subspace k, we measure its mean activation response as $A_k = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_k'^i\|_2$, where $\mathbf{x}_k'^i$ denotes the evolved representation of subspace k for token i, and N is the number of tokens. Figure 1 visualizes five representative DeiT-T S2E subspaces under the $K = 48$, $r = 16$ configuration, showing distinct activation distributions and spatial response patterns. These observations provide qualitative evidence of heterogeneous subspace behavior; however, the visual differences alone do not establish specific semantic interpretations or causal roles.

Subspace representation similarity. Pairwise cosine similarity among $K = 24$ representation groups in the last FFN block (Figure 2) compares the internal representation structure of the standard DeiT-T FFN and S2E. For the standard FFN baseline, the hidden representation is partitioned into contiguous groups matching the S2E subspace dimensionality solely for diagnostic comparison; this grouping does not modify the baseline architecture or inference procedure. The standard FFN exhibits higher mean off-diagonal similarity, indicating greater representational redundancy, whereas S2E reduces the mean off-diagonal cosine similarity from 0.8254 to 0.2828. This comparison provides evidence that S2E produces more separated representation groups within the shared hidden space.

To examine how subspace organization changes across network depth, we further visualize class-wise activation responses and pairwise subspace similarity at representative Transformer blocks in Figure 3. Under the DeiT-T S2E configuration with $K = 16$ and $r = 24$, the mean off-diagonal similarity varies substantially across depth, revealing a non-monotonic pattern. Early blocks exhibit stronger subspace separation, similarity increases at intermediate depth, and the final block shows renewed separation relative to the highly aligned intermediate block. These results indicate that subspace organization evolves across network depth rather than following a uniformly increasing or decreasing differentiation trend.

Quantitative representational diversity. We quantify representation separation using pairwise cosine dissimilarity $\delta_{ij} = 1 - \frac{\mathbf{x}_i'^\top \mathbf{x}_j'}{\|\mathbf{x}_i'\|_2 \|\mathbf{x}_j'\|_2}$ and its mean $\delta = \frac{2}{K(K-1)}\sum_{i<j}\delta_{ij}$. For comparison across architectures, diversity is computed between dimension-matched representation units. For

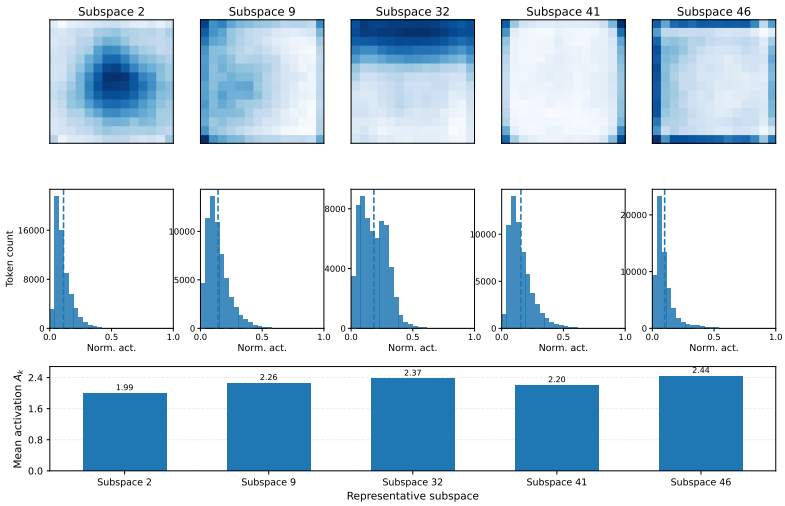
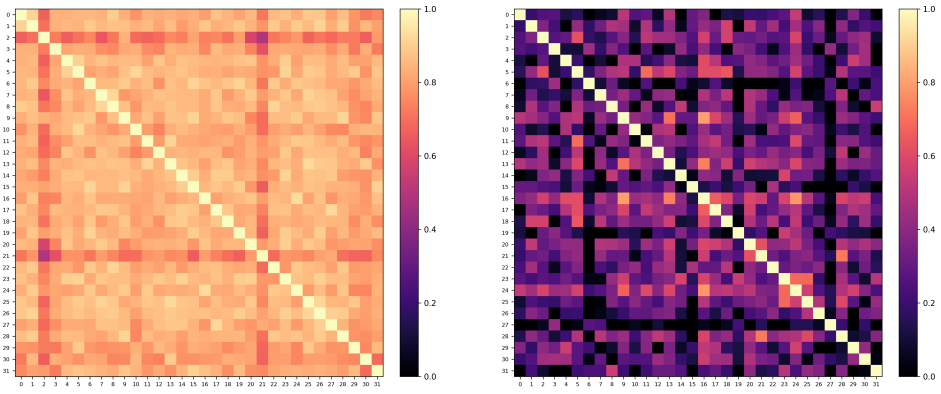


Figure 1: **Activation heterogeneity across representative S2E subspaces in DeiT-T ($K = 48$, $r = 16$).** Five representative subspaces are visualized using mean spatial activation maps, token-level normalized activation distributions, and mean activation magnitude A_k. The heterogeneous response patterns illustrate differentiated subspace behavior without implying specific semantic roles.



(a) Standard DeiT-T FFN (b) S2E

Figure 2: **Pairwise representation similarity in the standard DeiT-T FFN and S2E.** For the standard FFN, contiguous hidden-channel groups matching the S2E subspace dimensionality are used only for diagnostic comparison.

the dense FFN, the hidden representation is partitioned into contiguous, dimension-matched groups solely for diagnostic measurement; this grouping does not modify training or inference. Grouped FFN uses its native channel groups, standard MoE uses its native expert outputs, and S2E uses its learned semantic subspaces. Table 14 shows that S2E achieves higher representational diversity than the dense FFN, grouped FFN, and standard MoE baselines under this diagnostic measure. This indicates that structured organization of a shared representation can produce strong separation among subspace outputs without requiring independently parameterized experts.

Grouped-masking intervention. We further examine differentiated subspace contributions through grouped masking on DeiT-T. Subspaces are grouped according to their individual intervention effects, with high-impact subspaces corresponding to those associated with larger accuracy decreases under individual suppression and low-impact subspaces corresponding to those associated with smaller decreases. We additionally evaluate a randomly selected subspace group as a reference condition. As shown in Table 15, masking the high-impact group produces the largest reduction in Top-1 accuracy (-3.2 points), compared with low-impact masking (-2.0 points) and

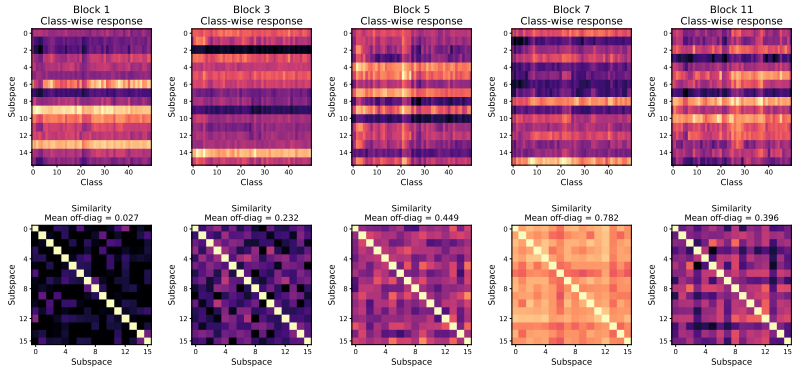


Figure 3: **S2E subspace behavior across representative Transformer blocks in DeiT-T** ($K = 16$, $r = 24$)**.** Top: class-wise activation responses over ImageNet categories; bottom: pairwise cosine similarity between subspaces. The results reveal a depth-dependent, non-monotonic pattern, with strong separation in early blocks, increased alignment at intermediate depth, and renewed separation in the final block.

Table 14: Representational diversity on DeiT-S. Diversity is computed between the native representation units of grouped FFN, MoE, and S2E; the dense FFN is partitioned into dimension-matched groups only for diagnostic evaluation. Higher δ indicates stronger representational separation.

Method	Architecture	Diversity $\delta \uparrow$
Dense FFN	Single representation	0.21
Grouped FFN	Fixed channel groups	0.34
Standard MoE	Independent experts	0.55
S2E (ours)	Shared semantic subspaces	**0.61**

random masking (-1.1 points). The stronger degradation produced by masking the high-impact group indicates that the learned subspaces do not contribute uniformly to prediction. Together with the representation-diversity and individual-subspace intervention analyses, these results provide additional evidence of differentiated functional behavior within the shared dense representation.

Table 15: Grouped-masking intervention on DeiT-T S2E. High- and low-impact groups are defined according to the magnitude of the accuracy decrease observed under individual-subspace suppression; random masking provides a reference condition.

Intervention	Top-1 Accuracy (%)	ΔTop-1
Original S2E	74.9	–
Random subspace masking	73.8	-1.1
High-impact subspace masking	71.7	-3.2
Low-impact subspace masking	72.9	-2.0

C Baseline Comparison on DeiT-T

This evaluation provides the full end-to-end comparison underlying Section 4.2. All end-to-end dense, grouped, and network-level MoE methods follow the DeiT evaluation protocol described in Appendix A, while post-hoc and factorized expert methods are included as structural references and identified separately. With the compact configuration (K=24, r=16, H=384), S2E achieves 74.9% Top-1 accuracy with 3.9M parameters and 0.90G FLOPs, improving over the FFN-4$\times$ baseline by $+2.7$ percentage points and over the matched-width FFN-2$\times$ baseline by $+3.4$ points. It also outperforms the evaluated network-level MoE variants while retaining fully dense computation. Under the hardware protocol described in Appendix A, this configuration achieves 41.4 ms batch latency, 6184 img/s throughput, and 13.5 GB peak memory. Compared with the matched-width FFN-2$\times$

baseline, S2E simultaneously improves accuracy and reduces measured latency, while relative to the evaluated MoE variants require substantially fewer parameters and lower peak memory and achieves higher throughput. Operator-level profiling further shows that centroid computation and similarity scoring account for 42% of the S2E block forward-pass time under the current implementation (Appendix B.6).

Table 16: Accuracy–efficiency comparison on DeiT-T, ImageNet-1K. Latency denotes mean batch latency at batch size 256, and throughput is computed as $\mathrm{Tp} = 256/(\mathrm{latency}(\mathrm{ms})/1000)$. Memory denotes peak GPU memory.

Method	Paradigm	Hidden Dim	Params (M)	FLOPs (G)	Top-1 (%)	Latency (ms)	Throughput (img/s)	Memory (GB)
Standard FFN-4×	Dense	768	5.7	1.26	72.2	49.7	5151	18.9
Standard FFN-2×	Dense	384	3.9	0.91	71.5	44.1	5805	12.9
RePaViT (Xu et al., 2025)	Dense	768	4.4	0.80	69.4	41.6	6154	13.6
ConvGLU (Shi, 2024)	Hybrid	768	5.9	1.40	74.2	45.5	5626	23.1
AFBO (Wu et al., 2025)	Dense	768	6.0	1.30	74.6	44.9	5702	22.3
IFFN (Xu et al., 2024)	Dense	768	5.0	1.10	72.6	44.5	5753	23.5
G-FFN (Li et al., 2022)	Grouped dense	768	5.2	1.05	72.8	52.2	4904	24.0
Standard MoE	Net-level sparse	768	23.4	1.96	69.9	65.0	3938	26.0
Soft MoE (Puigcerver et al., 2024)	Net-level soft	768	23.4	1.66	71.8	51.0	5020	24.2
AoE (Lv et al., 2025)	Router-free MoE	768	22.6	1.75	72.1	67.3	3804	23.5
MoEfication (Zhang et al., 2022)	Post-hoc	768	5.7	0.94	68.9	56.3	4547	21.6
MoEE (Berisha et al., 2025)	Post-hoc	768	4.5	0.95	69.7	55.6	4604	21.2
μMoE (Oldfield et al., 2024)	Factored expert	768	8.3	1.20	71.1	52.5	4876	22.1
S2E (K=24, r=16)	Subspace-level dense	384	3.9	0.90	**74.9**	41.4	6184	13.5
S2E (K=48, r=16)	Subspace-level dense	768	5.7	1.12	**75.8**	42.9	5967	19.0

D Generalization, Robustness, and Transfer

This appendix provides the complete evaluation supporting 4.2, including architectural generalization, distribution-shift robustness, and transfer to dense prediction tasks.

D.1 Full cross-backbone results

Table 17 evaluates the generality of S2E across diverse vision backbones and model scales. Across all evaluated architectures, S2E improves ImageNet-1K accuracy while reducing FLOPs and measured inference latency. Parameter reductions are also observed for all evaluated backbones except DeiT-T, for which S2E retains the same parameter count as the baseline. The largest clean-accuracy gains are obtained on DeiT-T ($+3.6$ points), PoolFormer-S12 ($+3.4$ points), and PVTv2-B1 ($+2.5$ points). Improvements are also observed across larger architectures, although their magnitude varies by backbone. Beyond clean accuracy, S2E consistently improves performance under distribution shift. Across the evaluated backbones, S2E lowers ImageNet-C mCE and improves accuracy on ImageNet-A, ImageNet-R, and ImageNet-Sketch. For example, on DeiT-T, mCE decreases from 71.1 to 66.2, while ImageNet-A, ImageNet-R, and ImageNet-Sketch accuracy increase from 7.3%, 32.6%, and 20.2% to 8.3%, 38.5%, and 23.1%, respectively. Similar improvements are observed across the remaining architectures, indicating that the gains of S2E extend beyond in-distribution ImageNet-1K performance. S2E also provides substantial computational-efficiency benefits. FLOPs and measured latency are reduced across all evaluated architectures, while parameter reductions are substantial for most backbones. Particularly large parameter reductions are observed for DeiT-B, Swin-B, and LV-ViT-M. These results show that S2E can improve predictive performance and distribution-shift robustness while simultaneously reducing computational cost, without relying on sparse routing or independently parameterized expert modules.

Overall, the cross-backbone evaluation indicates that S2E generalizes across the tested vision architectures and provides a favorable accuracy–robustness–efficiency trade-off across different backbone families and model scales.

D.2 Transfer to object detection and instance segmentation

To evaluate whether the benefits of S2E extend beyond image classification, we integrate S2E into Swin-T and Swin-S backbones for object detection and instance segmentation using Mask R-CNN with FPN. Both baseline and S2E models are trained with identical COCO 2017 protocols. Table 18 shows that S2E transfers effectively from image classification to COCO object detection and instance segmentation. On both Swin-T and Swin-S, it improves box and mask AP while reducing

Table 17: Vanilla backbones versus their S2E-enhanced variants under the configurations used for each cross-backbone evaluation; the DeiT-T S2E result uses $K = 48, r = 16$

Model	#Param. (M)	FLOPs (G)	Latency (ms)	IN-1k (%)	IN-C (↓)	IN-A (↑)	IN-R (↑)	IN-SK (↑)
DeiT-T	5.7	1.26	49.7	72.2	71.1	7.3	32.6	20.2
+S2E (ours)	5.7	1.12	42.9	**75.8**	**66.2**	**8.3**	**38.5**	**23.1**
PoolFormer-S12	11.9	1.8	27.2	77.2	69.8	7.0	37.7	25.4
+S2E (ours)	7.3	1.2	8.2	**80.6**	**65.0**	**8.9**	**42.5**	**27.6**
DeiT-S	22.1	4.3	52.1	79.8	54.6	19.8	41.9	29.4
+S2E (ours)	16.7	3.2	43.2	**81.9**	**53.3**	**20.8**	**45.3**	**31.9**
DeiT-B	86.6	16.9	57.9	81.8	48.5	27.4	44.9	32.0
+S2E (ours)	44.1	9.1	50.2	**82.7**	**45.3**	**31.1**	**47.4**	**33.3**
PVTv2-B1	14.0	2.1	49.5	78.7	65.1	14.6	41.7	28.9
+S2E (ours)	9.0	1.3	41.2	**81.2**	**63.6**	**16.7**	**44.1**	**31.1**
ViT-S	22.0	4.2	52.3	79.9	47.2	43.7	53.2	39.4
+S2E (ours)	16.9	2.6	42.8	**81.7**	**38.1**	**47.1**	**71.2**	**42.4**
Swin-T	28.3	4.5	51.9	81.2	62.0	21.7	41.3	29.0
+S2E (ours)	17.3	3.9	42.5	**83.2**	**56.4**	**26.0**	**45.8**	**31.7**
Swin-S	49.6	8.6	65.2	83.0	54.9	32.9	44.9	32.0
+S2E (ours)	32.2	6.1	54.9	**83.6**	**52.1**	**33.3**	**47.4**	**34.5**
Swin-B	87.8	15.2	61.2	83.5	54.4	35.8	46.6	32.4
+S2E (ours)	50.6	10.6	51.3	**83.8**	**51.5**	**36.4**	**49.0**	**34.0**
LV-ViT-S	26.3	4.8	49.7	81.4	53.3	36.3	45.2	31.5
+S2E (ours)	16.2	2.1	31.8	**82.1**	**52.0**	**38.3**	**48.2**	**36.2**
LV-ViT-M	55.8	10.3	54.2	83.6	54.5	36.5	45.1	33.2
+S2E (ours)	31.6	4.2	48.6	**83.8**	**49.3**	**40.2**	**51.2**	**39.3**

Table 18: Detection and instance segmentation on COCO 2017 val with Mask R-CNN, $1\times$ schedule. TP: throughput (img/s).

Backbone	Params (M)	FLOPs (G)	TP	AP_b	AP_m
ResNet50	44.2	246	2147	38.0	34.4
Swin-T	48.2	267	1809	43.7	39.8
Swin-T + S2E (ours)	39.9	196	2032	**46.3**	**40.5**
Swin-S	69.6	354	2243	49.9	41.6
Swin-S + S2E (ours)	58.3	214	2318	**51.7**	**42.9**

parameters and FLOPs. Swin-T improves from 43.7/39.8 to 46.3/40.5 AP, while Swin-S improves from 49.9/41.6 to 51.7/42.9 AP for box/mask prediction. Parameters and FLOPs are also substantially reduced, while throughput increases for both backbones. These results indicate that S2E's structured subspace organization generalizes to spatially structured prediction while maintaining favorable accuracy–efficiency characteristics.