SEMANTIC ROUTING IN PRE-TRAINED VISION MODELS FOR ONLINE DOMAIN-INCREMENTAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning in the real world requires models to evolve with changing environments and cope with diverse forms of distribution shift. This is especially challenging in online domain-incremental learning, where data arrive as a non-stationary stream, each sample can be seen only once, and past observations cannot be revisited. Although pre-trained models can be used to obtain strong initial representations, standard fine-tuning in this setting leads to forgetting and poor cross-domain generalization. Inspired by how the human brain organizes experiences around semantic concepts, we propose Semantic Adapters (SAD)—lightweight modules plugged on top of any frozen pre-trained vision encoder that leverage structured semantic knowledge to guide representation updates. By routing the updates toward semantic clusters rather than domains, SAD stabilizes learning while enabling fast, one-pass adaptation. To further enrich flexibility, we introduce SADLoRA, which augments heads with low-rank parameter updates within the encoder, further enhancing adaptability while maintaining efficiency. Extensive experiments across diverse domain shifts show that both SAD versions substantially reduces forgetting and accelerates adaptation. The proposed Semantic routing with targeted updation offers a simple, fast, scalable and a viable solution for robust continual adaptation in dynamic real-world scenarios.[1]

1 INTRODUCTION

Real-world applications need models that can continuously adapt as environments evolve, often facing diverse forms of distribution shift. Continual learning (CL) enables models to learn new tasks sequentially without forgetting previously acquired knowledge. A fundamental obstacle in CL is catastrophic forgetting (McCloskey & Cohen, 1989; French, 1999), reflecting the trade-off between stability (preserving past knowledge) and plasticity (adapting to new input). This challenge is commonly evaluated in Class- and Domain-incremental learning (IL) settings. While Class-IL introduces entirely new classes at each task, Domain-IL maintains a fixed label space but requires robustness to shifts in input distributions. The latter arises naturally in real-world scenarios, for example, an autonomous driving system must recognize the same traffic signs despite variations in geography, weather, lighting, or sensor modality. The online Domain-IL setting is especially demanding, as data arrive only once in a stream, replay is limited or absent, and updates must be fast and memory-efficient.

Learning in continual settings involves two tightly coupled aspects: shaping the underlying *representations* and adapting the *decision boundaries*, and forgetting can occur in both. To mitigate degradation in the representation space and leverage robust features, recent works increasingly rely on pre-trained or foundation models, which provide strong generalization and transfer capabilities. Recent works have begun exploring how these models can be adapted to CL settings through parameter-efficient strategies such as prompt tuning or rehearsal-based fine-tuning (Wang et al., 2022a; Zhou et al., 2025). These strategies can be effective when the incoming stream is close to the pretraining distribution, but under substantial domain shift they tend to overfit the latest domain and interfere with previously learned generic features, amplifying recency bias and brittle generalization (Borlino et al., 2024). Moreover, most studies with pre-trained models emphasize Class-IL; Domain-IL with such models—especially in the strict online regime—remains comparatively underexplored.

[1] Code will be made available upon acceptance.

Domain-incremental learning with pre-trained models presents distinct challenges: preserving generic, reusable representations while keeping the classifier unbiased. A common strategy is to freeze the encoder and update only lightweight adapters, which is computationally efficient. However, when these adapters are fine-tuned sequentially on a drifting stream, they still overwrite previously learned information and suffer from forgetting. Recent methods attach domain- or task-specific prompts or modules (Gao et al., 2024; Byun et al., 2024), which effectively capture each domain's distribution. Yet, such designs treat domains in isolation, chasing domain-specific characteristics rather than consolidating transferable knowledge. By contrast, humans organize experience around semantics: an "airplane" or "helicopter" remains an aerial vehicle whether seen as a photograph, a sketch, or a painting, or under different lighting and contexts. Cognitive studies suggest that knowledge is encoded in structured semantic networks that abstract away from surface variations (Binder et al., 2009; Ralph et al., 2017). This perspective motivates a shift from domain-centric adaptation toward meaning-centric updates, reducing interference and fostering reuse across heterogeneous shifts.

Building on this motivation, we propose **semantic routing** as a practical way to reduce forgetting and improve adaptability in continual learning. Instead of attaching domain-specific modules that quickly become brittle, our approach organizes learning around meaning: semantically related classes are grouped into coherent clusters, and inputs are routed to lightweight adapter heads that specialize within each cluster. This reduces cross-domain interference, simplifies decision boundaries, and naturally encourages knowledge reuse as new data arrive. To realize this idea, we rely on Language as it serves as a powerful, high-level abstraction prior to extract the semantic concepts in the visual inputs. The embeddings of class names or attributes from a pre-trained language model, are clustered into a handful of coherent and meaningful groups. For each cluster, we instantiate a **Semantic Adapter (SAD)** head—lightweight classifiers trained only on their member classes but shared across domains. The vision encoder remains frozen, and all learning happens in these adapter heads.

We further introduce **SAD-LoRA**, which augments SAD by inserting low-rank adapters into the last encoder blocks, combining semantic routing with targeted plasticity. Together, these modules provide a simple yet effective mechanism for continual adaptation that is computationally efficient. We evaluate semantic routing under one of the demanding continual learning scenarios: *online domain-incremental learning*, where data arrive in a single pass, memory and compute are constrained, and distribution shifts can be severe. Extensive empirical evaluation across multiple Domain-IL datasets and benchmarks demonstrates that the semantic approach consistently improves both final and anytime accuracy compared to multiple SOTA baselines, supporting the idea that organizing updates by meaning—rather than by domain—yields more stable online Domain-IL. We also show that such targeted semantic updating also proves more robust against natural corruptions or perturbations. Beyond empirical gains, the proposed semantic routing is a viable alternative to full fine-tuning, enabling practitioners to continually leverage pre-trained models to application-specific datasets in real-world environment.

2 Related Works

2.1 Continual Learning

Continual learning (CL) aims to enable the models to learn on a sequence of tasks without catastrophic forgetting. A common taxonomy distinguishes *task-incremental*, *class-incremental*, and *domain-incremental* scenarios depending on whether the output space or only the input distribution shifts across tasks (Van de Ven et al., 2022). In Domain–incremental learning (Domain-IL), the class label set is fixed across all stages while the input distribution changes. Several strategies are proposed to mitigate forgetting. Replay-based methods (Buzzega et al., 2020; Prabhu et al., 2020; Aljundi et al., 2019; Isele & Cosgun, 2018; Rolnick et al., 2019) address forgetting by storing a subset of past data for replay. ER-ACE couples rehearsal with class-balanced losses (anti-conflict) to mitigate bias toward recent classes (Caccia et al., 2021). Replay reduces drift but introduces storage/privacy costs and is less aligned with online single-pass settings. Regularization methods (Aljundi et al., 2018; Chaudhry et al., 2018) such oEWC assigns Fisher-based importance to parameters and penalizes their drift across tasks, implemented online with a running Fisher estimate (Kirkpatrick et al., 2017). SI accumulates path-integral importance that discourages changes to parameters that have contributed large loss reductions in the past (Zenke et al., 2017). Despite their simplicity, regularization methods often struggle when tasks are highly dissimilar, such as in domain-incremental learning settings

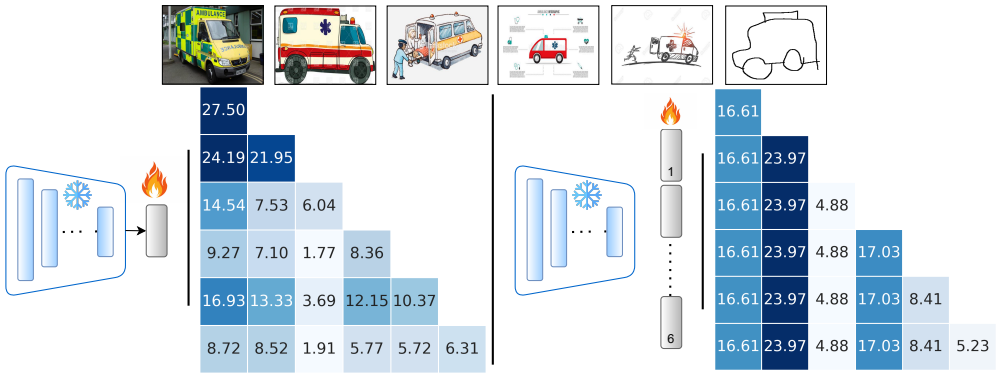

Figure 1: Online Domain-incremental learning using Pre-trained Vision Encoders on DN4IL dataset with six domains - Real, Clipart, Infograph, Painting, Sketch and Quickdraw

where input distributions shift substantially. Dynamic architectural methods (Mallya & Lazebnik, 2018; Rusu et al., 2016) tackle forgetting by expanding model capacity per task, freezing earlier weights, and learning new subnetworks.

2.2 Pre-trained Models-based Continual Learning

With the advent of large pre-trained models, many fine-tuning methods minimize forgetting by leveraging the pre-trained encoder and learning small plug-ins. L2P learns a pool of prompts and retrieves a subset per input and the prompt keys act as a router in feature space (Wang et al., 2022a). DualPrompt separates general and expert prompts and trains a prompt router for better transfer (Wang et al., 2022b). There are also fine-tuning based replay techniques. SimpleCIL revisits PTM-based CIL with a streamlined recipe (light distillation, cosine heads, balanced sampling), showing strong performance with a modest buffer but still relying on stored images (Zhou et al., 2025). MEMO shows if a small buffer can outperform a fine-tuned model without buffer and advocates memory-efficient rehearsal with careful sampling and tuning (Zhou et al., 2023). A common limitation is that many of these approaches operate in a *non-online regime*, requiring multiple epochs per domain, frequent full-batch retraining, or explicit task/domain ID information—not always feasible in truly streaming or resource-constrained settings. Moreover, prompt- or domain-specific adapters typically chase domain idiosyncrasies; when distributions keep changing, their parameters can themselves suffer from recency bias. Many works utilize pre-trained models but perform multi-stage training with full-retraining or model merging (Ainsworth et al., 2022; Matena & Raffel, 2022; Frankle et al., 2020). Our work aims to leverage the representations from a pre-trained model, and update only specific modules without updating the model.

3 Semantic Adapters

Our goal is to leverage rich representations from pre-trained vision models and perform efficient targeted adaptation in a domain-incremental setting. We begin with a simple empirical analysis on DN4IL to understand what a frozen pre-trained encoder can and cannot do under online domain shifts. We take an ImageNet-pre-trained ResNet-18, freeze all its weights, and only train classifier heads in a single pass over the six DN4IL domains (Real, Clipart, Infograph, Painting, Sketch, Quickdraw). After each stage we evaluate on all domains seen so far.

The left panel of Figure 1 trains a single linear classifier on top of the frozen backbone. The per-domain heatmap reveals severe forgetting—new domains overwrite the decision boundary for earlier ones. The right panel replaces this with domain-specific heads (requiring domain ID at test time). This removes cross-domain interference in the classifier, largely eliminating forgetting, however, the overall accuracy remains low on several domains. Each head relearns similar concepts independently (e.g., “ambulance” in Real vs. Sketch), failing to transfer structure such as shape or relations that recur across domains.

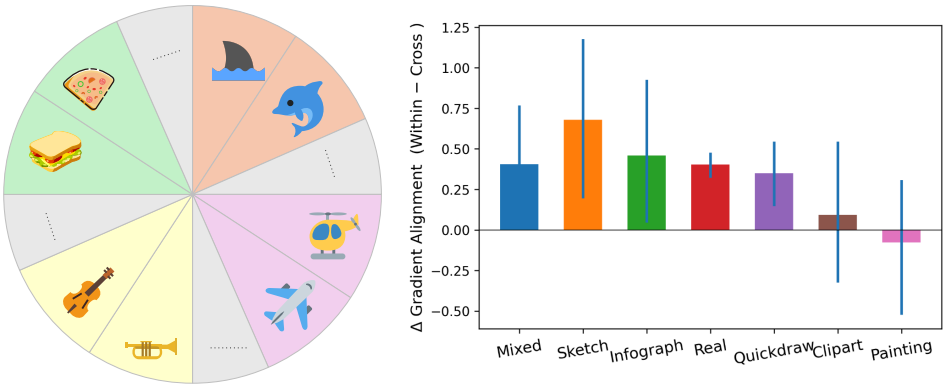

Figure 2: Semantic groups in DN4IL and Gradient Alignment within and outside the groups

Motivation Humans accumulate lifelong knowledge by organizing experiences into rich semantic abstractions rather than rote episodic traces. Classic studies of catastrophic interference show that naively updating a single network on new tasks rapidly erodes prior knowledge (McCloskey & Cohen, 1989; French, 1999). In contrast, semantic memory supports concept-level representations (e.g., "airplane," "helicopter," "hot-air balloon" as aerial vehicles; "pear," "apple," "strawberry" as fruits) that generalize across large visual changes(Tulving, 1972). Converging cognitive and neuroimaging evidence further indicates modular specialization—partially distinct subsystems tuned for faces, places, and objects—allowing stable representations with localized adaptation when new experience arrives (Kanwisher et al., 1997).

Guided by this view, we aim to replace domain-conditioned heads with semantic routing. We want to cluster the input data into a small set of concept groups and attach lightweight classifier heads to those groups. Inputs are routed by meaning, not by domain and can hence make use of the shared structure across domains.

Semantic Groups Analysis To test whether semantically related classes are indeed safer to share parameters, we conduct a gradient-alignment study. For the analysis we freeze a ResNet-18 encoder and train a small linear head on an anchor pair A (e.g., *airplane* vs. *helicopter*). At the resulting weights θ_A we compare gradients for two new pairs: a within-group pair B taken from the same semantic group as A (e.g., *airplane* vs. *helicopter*/*UFO*), and a cross-group pair C (e.g., *airplane* vs. *pizza*). We evaluate two settings: (i) per-domain, where images for A, B, C are sampled from a single domain (one task) to remove style variation; and (ii) mixed, where images are pooled across all domains, so each pair includes multi-style examples. For each domain/mode we summarize a compatibility score $\Delta\cos = \cos(g_A, g_B) - \cos(g_A, g_C)$, where positive values indicate that within-group updates push in more similar directions than cross-group updates.

Across DN4IL settings ("mixed" and "per-domain" slices), $\Delta\cos$ is consistently positive (Fig. 2, right) This means that, holding the backbone fixed, the optimization step needed for a semantically related pair is more aligned with the step required by the current task than the step for an unrelated pair. In other words, sharing parameters within a semantic group is cooperative, while sharing across groups induces conflicting updates. This provides direct motivation for our routing design: routing each input to its *semantic* adapter concentrates learning within a concept-consistent subspace, reducing interference and enabling targeted adaptation.

4 Methodology

We study Domain–Incremental Learning (Domain-IL), where the label space stays fixed while data domains change. Let $\mathcal{C} = \{c_1, \ldots, c_M\}$ be the shared class set. We train over a sequence of domains (tasks) $\{\mathcal{D}_b\}_{b=1}^{B}$, each inducing a different distribution over the same label space but with shifted input statistics. At task b, the training stream is $\mathcal{T}_b = \{(x_i^{(b)}, y_i^{(b)})\}_{i=1}^{N_b}$.

We use any frozen pre-trained vision encoder $f : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^D$ to encode the visual inputs to produce image embeddings. $z = f(x) \in \mathbb{R}^D$.

Language–Induced Semantic Grouping: At the beginning of the first task, we encode class labels or attributes using any light-weight pre-trained language model (PLM) and cluster them into semantic groups. This is to enable a meaningful, domain-agnostic structure on object relationships, enabling efficient routing and shared learning across heterogeneous domains.

At the beginning of Task0, we know all the classes that we will encounter in the whole training. Hence, for each class $c \in \mathcal{C}$, obtain a d-dimensional language embedding via a pre-trained language model (PLM),

$$e_c = \mathrm{PLM}(c) \in \mathbb{R}^d.$$

Clustering $\{e_c\}_{c\in\mathcal{C}}$ yields K disjoint semantic groups $\{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$ and a routing map

$$g : \mathcal{C} \to \{1, \ldots, K\}, \qquad g(c) = k \iff c \in \mathcal{S}_k.$$

For each group k, we also save a fixed language prototype

$$L_k \;=\; \mathrm{normalize}\left(\frac{1}{|\mathcal{S}_k|}\sum_{c\in\mathcal{S}_k} e_c\right) \in \mathbb{R}^d.$$

The semantic groups and language prototypes are computed once and remain fixed.

Semantic Adapters on Frozen Features: We enhance the frozen backbone with lightweight, group-specific linear classifier heads that focus on semantically coherent subsets of classes in the Domain-IL training. If there are K semantic clusters formed in the previous step, we instantiate K classifier heads with the corresponding classes per cluster. All the semantic adapter heads (classifiers) are instantiated once at the beginning of training. (Figure 3)

For each group k, the semantic adapter head is

$$A_k : \mathbb{R}^D \to \mathbb{R}^{|\mathcal{S}_k|}.$$

Given an input sample (x, y), we form $z = f(x)$ from the frozen encoder and train only the *matching* head $A_{g(y)}$:

$$f_{\text{sad}} \;=\; A_{g(y)}(z) \in \mathbb{R}^{|\mathcal{S}_{g(y)}|}.$$

Let $\mathrm{i}(y; \mathcal{S}_{g(y)})$ be the local index of y inside its group. The supervised loss on the routed head is the standard cross-entropy loss,

$$\mathcal{L}_{\mathrm{CE}} \;=\; -\log \frac{\exp\big([A_{g(y)}(z)]_{\mathrm{i}(y;\mathcal{S}_{g(y)})}\big)}{\sum_{c\in\mathcal{S}_{g(y)}} \exp\big([A_{g(y)}(z)]_{\mathrm{i}(c;\mathcal{S}_{g(y)})}\big)}.$$

Language Anchoring : Projecting visual features into the semantic language space and optimizing for alignment strengthens the semantic consistency between vision and language modalities, and proves useful for dynamic routing during inference. To encourage semantic alignment, we learn a small projection $P \in \mathbb{R}^{d\times D}$ that maps visual features to the PLM space, $\tilde{e} = \mathrm{normalize}(P(z)) \in \mathbb{R}^d$. We apply a group-level contrastive loss:

$$\mathcal{L}_{\text{lang\_anchor}} \;=\; -\log \frac{\exp\big(\langle \tilde{e}, L_{g(y)}\rangle/\tau_\ell\big)}{\sum_{k=1}^{K} \exp\big(\langle \tilde{e}, L_k\rangle/\tau_\ell\big)}.$$

Inter– and Intra-Group Separation: To further discourage interference across semantic subspaces and reduce redundancy within each head, we use a separation loss that penalizes off-diagonal cosine similarity. Let $\mathcal{B}$ be a mini-batch, and for each group k define the batch mean $\mu_k = \frac{1}{|\mathcal{B}_k|}\sum_{(x,y)\in\mathcal{B}_k} z$, where $\mathcal{B}_k = \{(x, y) \in \mathcal{B} : g(y) = k\}$, and $\hat{\mu}_k = \mu_k/\|\mu_k\|_2$.

$$\mathcal{L}_{\text{sep}} \;=\; \frac{1}{K(K-1)}\sum_{k\neq k'} \big\langle \hat{\mu}_k, \hat{\mu}_{k'} \big\rangle^2.$$

Similarly we also apply a similar *intra-group* de-correlation on classifier weights to make the classes inside each semantic head more distinct.

Overall Objective:

$$\mathcal{L} \;=\; \mathcal{L}_{\mathrm{CE}} \;+\; \lambda_{\text{lang}}\,\mathcal{L}_{\text{lang\_anchor}} \;+\; \lambda_{\text{sep}}\,\mathcal{L}_{\text{sep}}.$$

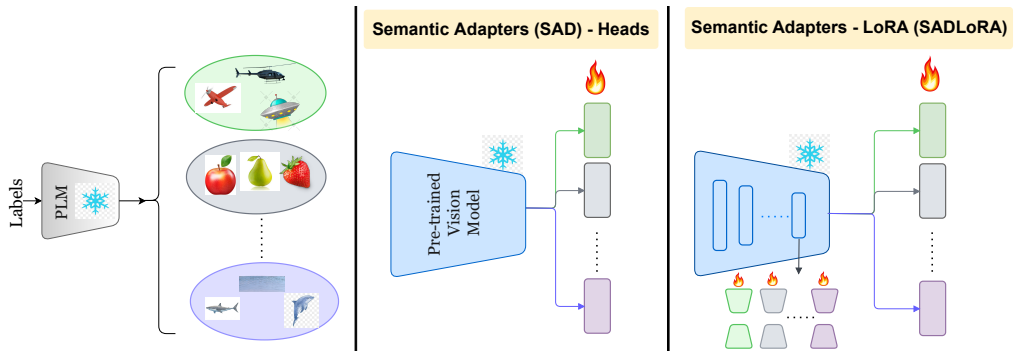

Figure 3: Semantic Adapters Heads (SAD) and Semantic Adapter LoRA (SADLoRA) for Pre-trained Vision encoders for Online Domain-incremental learning.

SAD–LoRA Beyond training the group heads, we optionally activate semantic LoRA adapters inside the frozen encoder for additional capacity without full fine-tuning. Let θ denote the backbone weights (frozen) and $\Delta\theta^{(k)}$ be group-specific low-rank updates (LoRA) inserted into a subset of layers:

$$W \;\mapsto\; W \;+\; A^{(k)}B^{(k)}, \qquad A^{(k)} \in \mathbb{R}^{m\times r},\; B^{(k)} \in \mathbb{R}^{r\times n},\; r \ll \min(m,n).$$

For ResNet-18 we attach LoRA to the final stage (layer4) convolutions; for ViT we attach LoRA to the last transformer block, targeting self-attention and MLP projections. We train only the group heads and the group-specific LoRA parameters in similar way as shown above.

Inference Strategies : Given a test sample x from domain b, set $z = f(x)$ and $\hat{z} = z/\|z\|_2$, and let $\tilde{e} = \text{normalize}(P(z))$.

Oracle uses the true group $k = g(y)$ (upper bound), to route the sample to the correct semantic adapter and predict the output.

Dynamic Routing: In real-world we need to dynamically route without oracle knowledge. At the beginning of training, when we do semantic grouping, we also save the language prototypes per group. Further, during training we also save the group-wise visual prototypes (running means of the features within that group). From the incoming sample, we compute a routing score using visual prototypes and language prototypes by passing it through the small projection layer, we trained. We also augment this with with a head-confidence term, energy or negative entropy of each head's logits. We use this score to select the active adapter and perform classification.

5 Experimental Setup

We begin our initial empirical analysis with the DN4IL dataset (Gowda et al., 2024), which is a curated subset of the DomainNet (Peng et al., 2019) dataset used in Domain Adaptation and considers the ease of benchmarking for continual learning purposes, and has siz domains/ We also benchmark on CORe50 (Lomonaco & Maltoni, 2017), which has eleven domains and Office-Home, which has four domains (Venkateswara et al., 2017). Unless stated, results are averaged over 3 seeds. We train with only 1 EPOCH to satisfy ONLINE Domain-incremental learning evaluation.

We use ImageNet-1K (Russakovsky et al., 2015) pre-trained ResNet-18 and ViT-B/16 as frozen feature extractors. The clusters and language prototypes are built with the CLIP text encoder (ViT-B/32;). We form K semantic groups per dataset by clustering class text features: DN4IL($K = 16$), OfficeHome ($K = 20$), CORe50($K = 10$). For example, in DN4Il - airplanes, helicopters, flying saucers classes get clustered together as they all can be categorized as "aerial vehicles". Other clusters characterized - fruts, indoor objects, outdoor scenes, clothing etc.

We report on multiple baselines, starting with few online CL methods - oEWC (Kirkpatrick et al., 2017), SI (Zenke et al., 2017) and exemplar based ER-ACE (Caccia et al., 2021). We also show

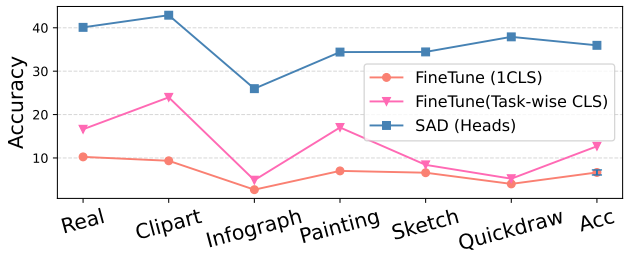

Figure 4: Accuracy on *DN4IL* dataset on Pre-trained ResNet18

Table 1: SAD results for Online Domain-IL on PTM-ResNet18 and OfficeHome and DN4IL datasets. Methods with † use replay.

	Method	DN4IL		OfficeHome		#Trainable Params (DN4IL / Off)
		Acc	*AAA*	*Acc*	*AAA*	
Scratch	SGD	6.32±1.86	6.23±0.48	4.03±0.61	3.69±0.22	11.23M / 11.21M
	oEWC	6.84±0.32	4.76±0.08	4.25±0.51	3.57±0.19	
	SI	7.14±0.72	6.26±0.32	4.61±0.80	3.43±0.54	
	ER-ACE†	5.12±0.49	4.85±0.51	3.11±0.93	3.55±0.52	
PTM	SGD	14.97±0.58	22.32±0.29	45.96±0.48	37.96±2.32	11.23M / 11.21M
	oEWC	12.54±1.35	17.95±0.59	48.84±2.71	41.29±2.40	
	SI	17.97±1.06	24.99±1.52	46.72±1.43	38.50±1.91	
	ER-ACE†	19.69±0.53	24.31±0.49	50.30±0.59	34.55±0.82	
PTM Frozen	SAD (oracle)	36.66±0.25	44.93±0.22	74.50±0.04	70.33±0.91	51K / 33K
	SAD	**20.45±0.30**	**26.09±0.30**	**60.32±0.30**	**57.71±0.42**	**313K / 296K**

comparison with explicit PTM-based CL methods, MEMO (Zhou et al., 2023), SimpleCIL (Zhou et al., 2025) and prompt-based methods L2P (Wang et al., 2022a), DualPrompt (Wang et al., 2022b). We also report multiple variants of our method : **SAD-Oracle**: evaluated with the oracle group-id; **SAD** : Semantic Heads evaluated with dynamic routing. We compare results with **SADLoRA** variant: SAD with *low-rank adapters* - at the end of the Results section.

Metrics Let T be the number of tasks and $a_{t\to k}$ denote the test accuracy on task k after completing training on task t. The per-step average accuracy is $A_t = \frac{1}{T}\sum_{k=1}^{T} a_{t\to k}$. We report Acc as average performance of all tasks after the last task and Averaged Anytime Accuracy (AAA) (Caccia et al., 2021) that evaluates the model through all tasks.

$$Acc = A_T \qquad \text{and} \qquad AAA = \frac{1}{T}\sum_{t=1}^{T} A_t \,.$$

6 Results

We begin with a simple analysis on DN4IL using a pre-trained ResNet18 model(Fig. 4). We compare three setups: FineTuning with a single head, FineTuning with domain–specific heads, and SAD which freezes the encoder and trains language–derived semantic heads shared across domains (Oracle mode). SAD shows higher per–domain accuracy and a stronger overall average, whereas fine–tuning suffers severe forgetting. Task–wise heads do not suffer forgetting, but has sub-optimal performance. Grouping classes by semantics encourages the model to reuse object–level cues across domains instead of overfitting to domain style, which is crucial under single–pass learning. The significant gap in performance in SAD-Oracle proves that if routed properly, the semantic modules can adapt well to improve plasticity on new tasks while also maintaining stablility over previous tasks.

Table 2: Comparison with PTM-based methods that run longer epochs. Methods with † use replay.

Method	Office-Home		CORe50		Epochs
	Acc	AAA	Acc	AAA	
MEMO† [ICLR'23]	63.1±1.80	71.2±2.76	68.2±2.7	66.0±2.7	20
L2P [CVPR'22]	80.0±1.29	79.7±4.19	81.7±0.4	72.3±1.1	10
DualPrompt [ECCV'22]	79.1±0.14	77.2±3.10	77.7±1.0	71.0±4.5	10
SimpleCIL [IJCV'24]	75.7±0.00	75.7±5.03	67.2±0.0	62.5±1.6	20
SAD-Oracle	88.2±0.13	85.6±0.21	90.1±0.3	86.7±0.16	1
SAD	80.1±0.08	77.8±0.8	84.8±0.56	81.1±0.46	1

Table 3: SAD vs. SADLoRA on Office-Home and CORe50

	Method	ResNet18				ViT-B/16			
		Office-Home		CORe50		Office-Home		CORe50	
		Acc	AAA	Acc	AAA	Acc	AAA	Acc	AAA
Oracle	SAD	74.50	70.33	72.74	67.81	88.21	85.63	90.10	86.70
	SADLoRA	81.20	73.12	81.66	78.51	89.24	86.38	91.85	87.46
Dynamic	SAD	60.32	57.71	60.35	53.59	80.14	77.83	84.82	81.06
	SADLoRA	69.62	64.21	72.52	67.45	81.80	78.16	85.78	82.09

Table 1 compares online Domain-IL on DN4IL and OfficeHome against few online CL baselines. We compare three training regimes: Scratch (randomly-initialized encoder), PTM (full fine-tuning of the ImageNet encoder), and PTM-Frozen (ours), which keeps the encoder fixed and trains only semantic heads. We compare against two regularization methods—oEWC and SI (which use importance-weighted regularization to slows weight drift)—and a replay method, ER-ACE (which does replay with asymmetric cross entropy to reduce bias). We report SAD (oracle)—an upper bound using the true semantic group for routing—and SAD with dynamic routing. While PTM baselines update the entire $11.2M$ parameter encoder each step, SAD trains only $300K$ parameters ($> 30x$ fewer) and yet attains competitive final accuracy on both datasets. Our semantic router and adapters can utilize the rich, transferable features from the frozen encoders, and only update specialized adapters—preserving general features, limiting forgetting, and letting each head specialize to its slice of the space.

Table 2 extends the study to ViT-B/16 architecture on OfficeHome and CORe-50 datasets. Here, we compare against methods specifically proposed for PTM-based CL, although they are not online and were proposed for Class-IL settings. We report the exemplar based MEMO, adapter based SimpleCIL and prompt-based L2P and DualPrompt. These methods all train the PTM for 10–20 epochs and depend on extra resources (buffers or prompt pools) and some also retrain the whole model. SAD is competitive or better on both the datasets and in the stricter 1 epoch regime. Notably, our model updates $290K$ trainable parameters versus $82M$ parameters in the backbone. Hence the frozen PTM features + semantically routed heads yield strong generalization and proves that targeted plasticity is better than global update by fine-tuning the whole model.

Adding low-rank plasticity. Table 3 evaluates SAD-LoRA, which introduces low-rank updates in the encoder tail along with the semantic heads. Across Office-Home and CORe50 with both ResNet-18 and ViT-B/16, SAD-LoRA consistently improves both Oracle and Dynamic settings, with a modest increase in trainable parameters (Table 5), still much lesser than retraining any encoder. The gains are consistently larger on ResNet-18 than on ViT-B/16 as placing LoRA in the final blocks aligns especially well with CNNs, where late layers carry task-specific cues. SAD-LoRA concentrates updates where they fix domain shift without destabilizing earlier, reusable features, yielding bigger returns for the CNN architecture while still giving consistent, smaller boosts for transformers. When domain shift is large, a small amount of targeted backbone plasticity complements the semantic heads and further stabilizes performance across the stream.

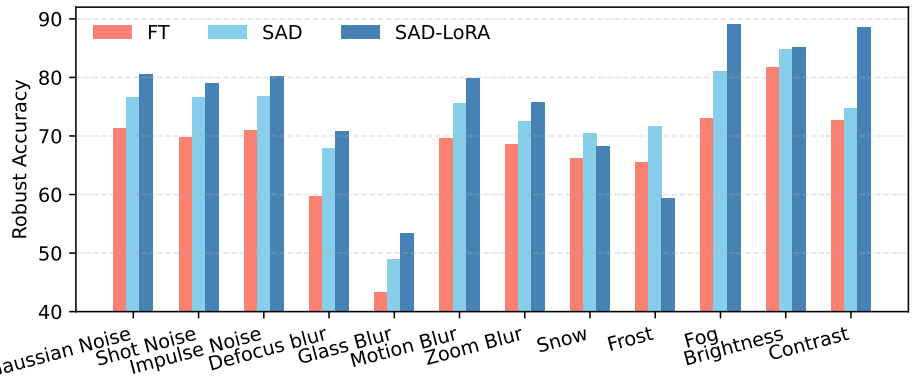

Figure 5: Robustness against different natural corruptions on ViT-B/16 and Office-home dataset

Robustness Analysis To assess deployment readiness under real-world degradations, we evaluate robustness to natural corruptions on Office-Home . We apply 12 corruption types (Hendrycks & Dietterich, 2019) at severity 3 to the "real" domain test set. Across all corruptions (Figure 5), SAD consistently outperforms full fine-tuning, and SAD-LoRA yields the strongest results—especially on blur, fog, and contrast—indicating that a small, targeted capacity inside the backbone helps recondition features without destabilizing shared representations. Semantic concepts act as high-level anchors: by routing features through concept-aligned heads, the model bases decisions on stable structure (shape/parts/relations) rather than fragile pixel cues, making it less sensitive to noise or blur.

Computational Efficiency Our goal was to keep the backbone encoder (CNN or Transformer) almost entirely frozen during continual learning, training only lightweight semantic heads and—under SAD-LoRA—low-rank adapters in the tail. Table 5 details the budget across datasets/architectures. For example, on the Office-Home dataset using ResNet-18, all semantic heads collectively have around 33,345 parameter, plus an additional projection layer of 262,656 parameters used for aligning vision and language embeddings to route inputs dynamically. This brings the total trainable parameters to roughly 296,000, which is minuscule compared to baseline methods that retrain the entire backbone, which have over 11 million parameters. Notably, our approach aso completes training of heads and adapters in just one epoch per domain, enabling rapid and resource-efficient online adaptation without sacrificing accuracy, making it especially suitable for quick adaptation.

7 Conclusion

We address online Domain-incremental learning setting, where non-stationary data streams demand rapid adaptation without replay and with tight compute/memory budgets. Leveraging rich representations from pre-trained encoders is appealing, but naive fine-tuning tends to overfit new domains and erode previously learned structure. Our goal was to gain targeted plasticity for the current domain while preserving stable, reusable features from the pre-trained models. We introduced semantic routing : language-induced clusters define concept groups, and we train lightweight, group-specific adapter heads on frozen encoders. We further proposed SAD-LoRA, which adds low-rank updates only in the encoder tail to provide more targeted plasticity without sacrificing stability. Across DN4IL, Office-Home, and CORe50 datasets, SAD/SAD-LoRA exceed strong PTM-based CL baselines while training only hundreds of thousands of parameters (vs. tens of millions for full fine-tuning). These results indicate that routing by meaning reduces gradient cross-talk, while targeted plasticity confines updates to the appropriate semantic subspace, preserving generic features that transfer across domains. Semantic routing also proves robust against natural corruptions. By aligning adaptation with human-meaningful semantics and keeping updates lightweight, our approach offers a practical, resource-efficient path to continual learning under real-world distribution shifts. As a broader impact, for practitioners seeking continual adaptation of pre-trained models on application-specific datasets, SAD provide a viable alternative to full fine-tuning.

REFERENCES

Samuel Ainsworth, Jonathan Hayase, and Siddhartha Srinivasa. Git re-basin: Merging models modulo permutation symmetries. In *The Eleventh International Conference on Learning Representations*, 2022.

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 139–154, 2018.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

Jeffrey R Binder, Rutvik H Desai, William W Graves, and Lisa L Conant. Where is the semantic system? a critical review and meta-analysis of 120 functional neuroimaging studies. *Cerebral cortex*, 19(12):2767–2796, 2009.

Francesco Cappio Borlino, Lorenzo Lu, and Tatiana Tommasi. Foundation models and fine-tuning: A benchmark for out of distribution detection. *IEEE Access*, 12:79401–79414, 2024.

Pietro Buzzega, Matteo Boschini, Andrea Porrello, and Simone Calderara. Der++: Improved deep episodic replay for continual learning. In *CVPR*, 2020.

Yewon Byun, Sanket Vaibhav Mehta, Saurabh Garg, Emma Strubell, Michael Oberst, Bryan Wilder, and Zachary C Lipton. Generate to discriminate: Expert routing for continual learning. *CoRR*, 2024.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *International Conference on Learning Representations*, 2021.

Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 532–547, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3 (4):128–135, 1999.

Zhanxin Gao, Jun Cen, and Xiaobin Chang. Consistent prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28463–28473, 2024.

Shruthi Gowda, Bahram Zonooz, and Elahe Arani. Dual cognitive architecture:: Incorporating biases and multi-memory systems for lifelong learning. *Transactions on Machine Learning Research*, 20 (X), 2024.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Nancy Kanwisher, Josh McDermott, and Marvin M Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *Journal of Neuroscience*, 17(11):4302–4311, 1997.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, and et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2017.

Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on robot learning*, pp. 17–26. PMLR, 2017.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, 2018.

Michael S Matena and Colin A Raffel. Merging models with fisher-weighted averaging. *Advances in Neural Information Processing Systems*, 35:17703–17716, 2022.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of learning and motivation*, 24:109–165, 1989.

Xingchao Peng, Qing Bai, Xiang Xia, Zuxuan Huang, and Kate Saenko. Moment matching for multi-source domain adaptation. In *ICCV*, 2019.

Ameya Prabhu, Philip HS Torr, and Puneet Kumar Dokania. Gdumb: A simple approach that questions our progress in continual learning. *ECCV*, 2020.

Matthew A Lambon Ralph, Elizabeth Jefferies, Karalyn Patterson, and Timothy T Rogers. The neural and computational bases of semantic cognition. *Nature reviews neuroscience*, 18(1):42–55, 2017.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. In *arXiv preprint arXiv:1606.04671*, 2016.

Endel Tulving. *Episodic and semantic memory*. Academic Press, 1972.

Gido M Van de Ven, Tinne Tuytelaars, and Andreas S Tolias. Three types of incremental learning. *Nature Machine Intelligence*, 4(12):1185–1197, 2022.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5018–5027, 2017.

Xialei Wang, Michael U. Gutmann, and Mark O'Neill. Learning to prompt for continual learning. In *CVPR*, 2022a.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pp. 631–648. Springer, 2022b.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Synaptic intelligence: Towards learning without forgetting. *arXiv preprint arXiv:1703.04200*, 2017.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133(3):1012–1032, 2025.

A Appendix

B Experimental Details

Benchmarks and Streams We study online Domain–IL on three standard vision benchmarks. DN4IL (Gowda et al., 2024) is a curated subset of DomainNet (Peng et al., 2019) containing six domains (Real, Clipart, Infograph, Painting, Sketch, Quickdraw) under a fixed class set. Office-Home (Venkateswara et al., 2017) has four domains (Art, Clipart, Product, Real-World), and CORe50 (Lomonaco & Maltoni, 2017) has eleven sessions commonly used as domains. Each dataset is processed as a single-pass stream: images arrive once per domain, we shuffle within a domain, and move to the next domain without revisiting previous data (unless a baseline explicitly requires replay). DN4IL images are 64×64. All other images are of 224×224 and normalized with ImageNet statistics. Results are averaged over three seeds.

Language-Induced Semantic Groups We construct meaning-centric groups once at the beginning of training. We encode class names with the CLIP text encoder (ViT-B/32). We also tried with class-level descriptions or attributes of each object, Example - "airplane" - a vehicle that has wings and engines and is capable of moving through the air. These light descriptions with attributes gave more nuances clusters. Text features are ℓ_2-normalized and clustered with k-means to form K disjoint semantic groups. We obtain $K{=}16$ for DN4IL, $K{=}20$ for Office-Home, and $K{=}10$ for CORe50. The language anchor for a group is the mean of its members' text embeddings, normalized. Group assignments and anchors remain fixed for the entire run.

Architectures and Trainable Modules We use ImageNet-1K pre-trained **ResNet-18** and **ViT-B/16** as *frozen* encoders. On top, we attach one *Semantic Adapter* per group: a linear classifier from the visual feature space ($D{=}512$ for ResNet-18, $D{=}768$ for ViT-B/16) to the classes in that group. In addition, we train a lightweight *vision→language* projection $W \in \mathbb{R}^{512\times D}$ to align visual features with language space (parameter counts match: $262{,}656$ for ResNet-18, $393{,}728$ for ViT-B/16). In **SAD-LoRA**, we further activate low-rank adapters in the encoder tail: rank $r{=}8$, $\alpha{=}16$, dropout 0.0. For ResNet-18 we instrument `layer4` convolutions; for ViT-B/16 we instrument the last transformer block (QKV and MLP projections). All remaining encoder weights stay frozen.

Training and Losses Unless noted, each stream is trained for **one epoch** (online). Batch size is 32. We optimize only the semantic heads, the vision→language projection, and (when enabled) LoRA parameters. For ResNet-18 we use SGD (lr 0.1); for ViT-B/16 we use AdamW (lr 0.001). The objective combines: (i) cross-entropy on the routed head; (ii) a language anchoring term that encourages the projected visual feature to score highest on its group anchor; and (iii) a group separation penalty that discourages cosine similarity between inter and intra groups. We set $\lambda_{\text{lang}}{=}0.5$ and $\lambda_{\text{sep}}{=}0.1$ and keep them fixed across datasets.

Dynamic Inference and Calibration At inference, we compute scores from two branches and fuse them: (1) *visual* scores via cosine similarity to vision prototypes; (2) *language* scores via the projected feature against text prototypes. Unless specified in the ablation, the encoder remains frozen and routing is fully dynamic.

Baselines and Their Resources We compare with two families. *Online CL baselines* include oEWC (Kirkpatrick et al., 2017) (online Fisher-based regularization) and SI (Zenke et al., 2017) (path-integral importance), both without replay, plus ER-ACE (Caccia et al., 2021) which uses rehearsal with asymmetric cross-entropy to reduce bias. *PTM-based CL baselines* include MEMO (Zhou et al., 2023) (memory-efficient rehearsal), L2P (Wang et al., 2022a) (prompt-pool retrieval for ViT), DualPrompt (Wang et al., 2022b) (general+task prompts with routing), and **SimpleCIL** (Zhou et al., 2025) (streamlined PTM fine-tuning with rehearsal). For methods with buffers or prompt pools, we follow the settings in their papers; when they train for multiple epochs (10–20), we report their regime while our default remains single-epoch online.

Metrics and Reporting We report **Acc** (final average over all tasks after the last domain) and AAA (Averaged Anytime Accuracy, i.e., the mean of per-time average accuracies across the stream).

Table 4: Accuracy on the *DN4IL* dataset on Pre-trained ResNet18 vision encoder.

Method	real	clipart	infograph	painting	sketch	quickdraw	Acc
	PTM						
FineTune + 1CLS	10.25	9.35	2.69	7.02	6.60	4.02	6.66]
FineTune + Task-Classifiers	16.61	23.97	4.88	17.03	8.41	5.22	12.69
	PTM + Semantic Adapters + Oracle-ID						
SAD	41.37	43.26	27.57	35.56	35.72	36.45	**36.66**
SADLoRA	55.36	54.94	29.55	47.80	48.63	65.11	**50.24**

Table 5: **Trainable parameter counts** for each architecture–dataset pair under our two variants. SAD trains only the semantic *classifiers* and the vision–language *projection* head. SAD-LoRA additionally trains the LoRA components inserted in the backbone tail. Counts are reported as absolute numbers

Dataset	Arch	Variant	Total trainable	Classifiers	Proj	LoRA
DN4IL	ResNet-18	SAD	313,956	51,300	262,656	–
DN4IL	ResNet-18	SAD-LoRA	4,008,548	51,300	262,656	3,694,592
Office-Home	ResNet-18	SAD	296,001	33,345	262,656	–
Office-Home	ResNet-18	SAD-LoRA	3,654,721	33,345	262,656	3,358,720
CORe50	ResNet-18	SAD	288,306	25,650	262,656	—
CORe50	ResNet-18	SAD-LoRA	1,967,666	25,650	262,656	1,679,360
Office-Home	ViT-B/16	SAD	443,713	49,985	393,728	–
Office-Home	ViT-B/16	SAD-LoRA	1,185,593	49,985	393,728	741,888
CORe50	ViT-B/16	SAD	432,178	38,450	393,728	–
CORe50	ViT-B/16	SAD-LoRA	805,426	38,450	393,728	373,248

We repeat each experiment with three random seeds and report mean $\pm$ std. Implementation is in PyTorch. Parameter counts in the main paper and the appendix table break down trainables into classifiers, projection head, and LoRA components for full transparency.

C Semantic Groups

We use a frozen pre-trained vision encoder (ResNet-18 or ViT-B/16) and attach lightweight *semantic adapter heads*. To define which classes share an adapter, we build *language-driven* groups: each class label is embedded with a pre-trained language model (e.g., CLIP-text or a sentence encoder), the label embeddings are Ł_2–normalized and clustered with k-means, and the resulting clusters are treated as semantically coherent groups. The group centroid serves as a fixed language prototype used for routing/analysis; only the small adapter heads are trained online. This grouping is computed once per dataset and remains fixed throughout training, ensuring a domain-agnostic, concept-level structure that encourages knowledge sharing across domains while keeping the backbone frozen.

DN4IL

1. Air Vehicles: airplane, helicopter, flying_saucer, hot_air_balloon
2. Vehicles: aircraft_carrier, bicycle, bus, motorbike, pickup_truck, train
3. Fruits and veggies: apple, carrot, onion, pear, strawberry, watermelon, banana, asparagus, broccoli
4. Hand Tools: axe, eraser, hammer, pencil, saw, scissors, screwdriver
5. Desserts & Fast Food: birthday_cake, ice_cream, pizza, sandwich, hamburger
6. Animals: bat, bird, dolphin, fish, mouse, rabbit, raccoon, squirrel, whale
7. Animals aquatic): dolphin, fish, shark, whale

8. MAMMALS: bear, camel, cow, dog, elephant, horse, kangaroo, lion, panda, tiger, zebra
9. INDOOR FURNITURE: bed, chair, couch, dresser, keyboard, table
10. INSECTS / ARACHNIDS: bee, butterfly, mosquito, spider
11. CLOTHING: bowtie, jacket, pants, shorts, sock
12. BUILDINGS: bridge, castle, house, skyscraper, windmill
13. NATURE / SCENES: bush, cloud, mountain, mushroom, ocean, river
14. HOUSEHOLD & MISC.: calendar, clock, cup, floor_lamp, frying_pan, map, marker, teapot, telephone, television, wine_bottle, wine_glass
15. MUSICAL INSTRUMENTS: cello, clarinet, guitar, trombone, violin
16. SEA : crab, lobster, octopus, scorpion, snail

OFFICE-HOME

1. WRITING TOOLS: Eraser, Pencil, Pen, Marker, Push_Pin
2. STORAGE: Shelf, File_Cabinet
3. ELECTRONICS / MEDIA: Alarm_Clock, Speaker, Radio, Webcam, Fan, Printer, TV, Monitor
4. FURNITURE: Couch, Table, Bed, Chair
5. FOOTWEAR: Sneakers, Flipflops
6. KITCHEN & TOOLS: Spoon, Knives, Paper_Clip, Scissors, Fork, Ruler, Pan, Hammer
7. OFFICE SUPPLIES: Calendar, Clipboards, Backpack, Notebook, Postit_Notes, Folder
8. POWER TOOLS: Screwdriver, Drill
9. SIGNAGE: Exit_Sign
10. COMPUTING: Computer, Keyboard, Laptop
11. TRANSPORT: Bike
12. CLEANING / HYGIENE: Mop, ToothBrush
13. CONTAINERS & MISC.: Mug, Bottle, Helmet, Soda, Trash_Can, Bucket
14. EYEWEAR: Glasses
15. OFFICE ELECTRONICS: Calculator, Telephone, Mouse
16. KITCHEN FIXTURES: Kettle, Sink
17. LIGHTING: Desk_Lamp
18. DECOR / MISC.: Flowers, Toys, Batteries
19. APPLIANCES: Refrigerator, Oven
20. HOME DECOR: Lamp_Shade, Candles, Curtains

D LARGE LANGUAGE MODELS USAGE IN WRITING PAPER

We used LLms to find potentially relevant prior work so we could double-check coverage of all the baselines and works related to our work. We also used it to reorganize paragraphs, improve clarity and grammar, tighten phrasing, and standardize tone (including some figure/table captions).