Free-Grained Hierarchical Visual Recognition

Supplementary Material

Contents

A. ImageNet-3L Dataset Construction **12**

B. Complete Hierarchy of ImageNet-3L **12**

C. Experiments on Messy Hierarchy **18**

D. More Experimantal Results **20**
D.1. Evaluaion on CUB-F 20
D.2. Evaluation under Varying and Severe Label Sparsity Conditions 20

E. t-SNE Visualization **22**

F. Ablation Study **23**
F.1. Importance of Text-guided Pseudo Attributes 23
F.2. Combining Text-Attr and Taxon-SSL 23
F.3. Ablation on the Text Encoder in Text-Attr 24
F.4. Ablation on Hierarchical Supervision in ViT 24
F.5. Ablation on Captioning Strategy and Caption Cost 25

G. Implementation Details **26**

H. Full Losses for Taxonomy-guided Semi-Supervised Learning (Taxon-SSL) **27**

I. Related Work **28**

A. ImageNet-3L Dataset Construction

Our hierarchy design is inspired by cognitive studies [31], which identify the *basic level* (e.g., *dog*) as the most natural and informative category for human recognition. Motivated by this, we construct a three-level hierarchy (*basic*, *subordinate*, and *fine-grained*), starting from this most informative level and avoiding overly abstract categories (e.g., *entity*, *artifact*) that are less meaningful for visual recognition.

Importantly, the notion of the basic level is not fixed. As shown in [31], category assignments can vary depending on context (e.g., "*fish*" may appear at superordinate or basic depending on the experiment), suggesting that level boundaries are inherently flexible. We therefore adopt Rosch's taxonomy as a guideline rather than a strict definition.

1. Defining the Basic Level: We define the *basic level* as categories that are both semantically meaningful and visually informative, ensuring comparisons at a consistent granularity. We primarily adopt categories aligned with Rosch's superordinate level (e.g., *bird*, *musical instrument*), which better match the scale of ImageNet, while avoiding overly coarse concepts (e.g., *entity*, *animal*) and overly fine-grained ones. For example, we prefer *dog*, *bird*, and *snake* over broader categories such as *mammal* or *reptile*.

Classes not covered by Rosch's taxonomy are mapped to WordNet nodes at comparable semantic depths, and categories that are too coarse or too fine are adjusted to nearby levels. The resulting basic level is fixed globally, and all nodes above it are removed to enforce a uniform starting point. This avoids mismatched comparisons, such as between fine-grained label (e.g., *teddy bear*) and coarse label (e.g., *conveyance*), and ensures semantically coherent grouping across levels.

2. Enforcing a Multi-level Hierarchy: We retain only categories that form meaningful three-level structures from *basic* to *subordinate* to *fine-grained*. Categories that collapse into a single path are removed, including cases where nodes have only one child at each level or where the hierarchy terminates early. For example, if a basic-level category leads to only one subordinate class, which further has only one fine-grained class, the hierarchy provides no meaningful distinction across levels. Similarly, we exclude shallow structures where the hierarchy does not extend to all three levels after defining the basic level. These pruning steps ensure that each retained category supports non-trivial branching and meaningful differentiation across levels.

3. Selecting Categories for Diversity: When multiple valid candidates exist at a given level of the taxonomy, we select the category that leads to the richest set of descendant classes. This encourages greater intra-group diversity and supports more meaningful distinctions at finer levels. For example, under *bird*, both *parrot* and *cockatoo* are valid candidates. However, *cockatoo* leads to only a single fine-grained class (e.g., *sulphur-crested cockatoo*), whereas *parrot* covers multiple diverse species (e.g., *African grey*, *sulphur-crested cockatoo*). We therefore select *parrot* to ensure broader coverage and richer fine-grained classification.

4. Handling Ambiguities and Naming: While WordNet provides a structured hierarchy, some categories are ambiguous or inconsistently defined. In such cases, we reorganize them using semantically coherent groupings. For example, instead of ill-defined categories such as *Women's Clothing*, we restructure them into functional groups (e.g., *Underwear*) to improve clarity and consistency.

5. Quality Control: We ensure taxonomy quality through a rule-based, human-in-the-loop process that verifies parent–child consistency and sibling-level coherence throughout construction. We further validate the structure using AI-assisted review (e.g., ChatGPT [1]) to identify potential inconsistencies or violations of intuitive categorization, followed by manual verification.

B. Complete Hierarchy of ImageNet-3L

Basic	**Subordinate**	**Fine-Grained**
bird	passerine bird	brambling, indigo bunting, robin, jay, bulbul, water ouzel, house finch, chickadee, junco, magpie, goldfinch
	parrot	macaw, sulphur-crested cockatoo, African grey, lorikeet
	piciform bird	toucan, jacamar
	seabird	king penguin, pelican, albatross
	anseriform bird	drake, red-breasted merganser, black swan, goose

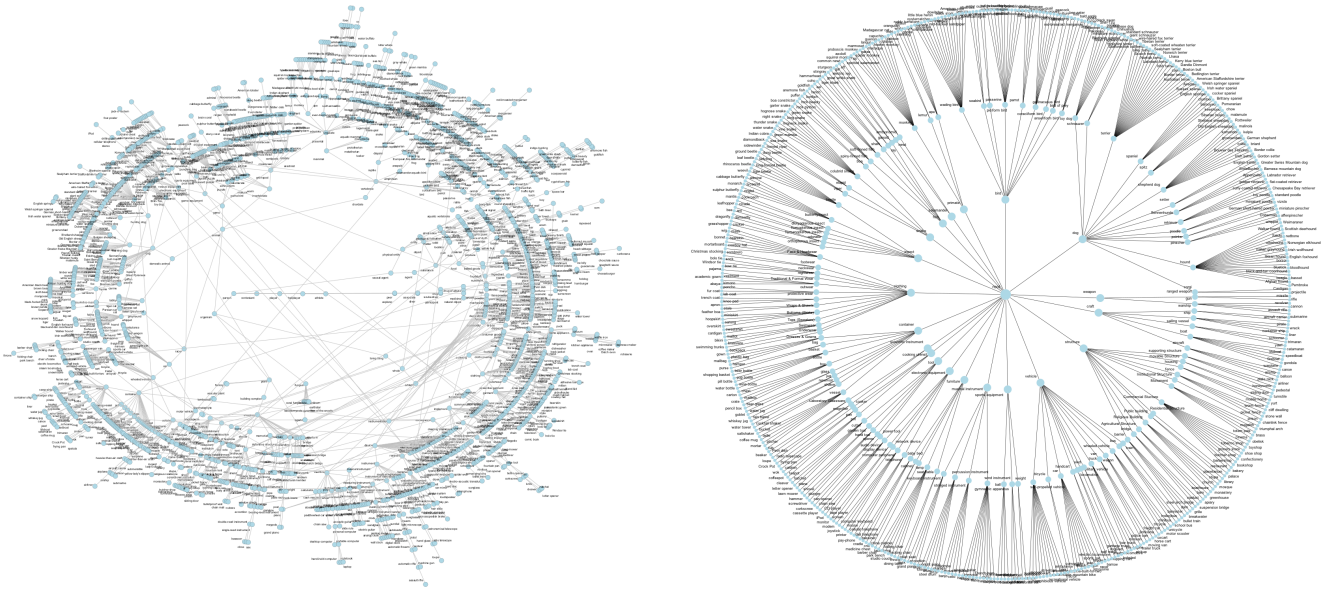
	coraciiform bird	bee eater, hornbill
	bird of prey	kite, great grey owl, vulture, bald eagle
	gallinaceous bird	partridge, prairie chicken, ruffed grouse, peacock, quail, black grouse, ptarmigan
	wading bird	flamingo, American coot, redshank, American egret, little blue heron, white stork, limpkin, spoonbill, red-backed sandpiper, dowitcher, crane, ruddy turnstone, bittern, oystercatcher, black stork, bustard
dog	spitz dog	malamute, Pomeranian, keeshond, Siberian husky, chow, Samoyed
	pointer dog	vizsla, German short-haired pointer
	spaniel dog	Brittany spaniel, clumber, English springer, Sussex spaniel, Irish water spaniel, Welsh springer spaniel, cocker spaniel
	hound dog	basset, bloodhound, Irish wolfhound, Walker hound, redbone, English foxhound, Italian greyhound, Ibizan hound, bluetick, Scottish deerhound, borzoi, Norwegian elkhound, whippet, Weimaraner, Saluki, beagle, Afghan hound, black-and-tan coonhound, otterhound
	terrier dog	Boston bull, silky terrier, Lakeland terrier, Yorkshire terrier, Tibetan terrier, American Staffordshire terrier, Irish terrier, Airedale, Norwich terrier, soft-coated wheaten terrier, wire-haired fox terrier, Staffordshire bullterrier, West Highland white terrier, Australian terrier, Dandie Dinmont, Kerry blue terrier, Lhasa, cairn, Sealyham terrier, Bedlington terrier, Scotch terrier, Border terrier, Norfolk terrier
	corgi dog	Pembroke, Cardigan
	poodle dog	miniature poodle, toy poodle, standard poodle
	setter dog	Irish setter, Gordon setter, English setter
	pinscher dog	Doberman, affenpinscher, miniature pinscher
	shepherd dog	kelpie, briard, German shepherd, Old English sheepdog, Border collie, Bouvier des Flandres, collie, Rottweiler, komondor, malinois, groenendael, Shetland sheepdog
	retriever dog	curly-coated retriever, Labrador retriever, Chesapeake Bay retriever, flat-coated retriever, golden retriever
	schnauzer dog	standard schnauzer, miniature schnauzer, giant schnauzer
	Sennenhunde dog	Bernese mountain dog, Greater Swiss Mountain dog, Appenzeller, EntleBucher
	toy dog	toy terrier, Blenheim spaniel, Maltese dog, Shih-Tzu, papillon, Pekinese, Chihuahua, Japanese spaniel
fish	soft-finned fish	coho, tench, eel, goldfish
	shark	tiger shark, great white shark, hammerhead
	spiny-finned fish	anemone fish, puffer, lionfish, rock beauty
	ray	stingray, electric ray

	ganoid fish	sturgeon, gar
primate	ape	gibbon, siamang, orangutan, chimpanzee, gorilla
	monkey	titi, langur, colobus, squirrel monkey, baboon, guenon, marmoset, macaque, spider monkey, patas, howler monkey, proboscis monkey, capuchin
	lemur	Madagascar cat, indri
snake	colubrid snake	water snake, garter snake, green snake, night snake, hognose snake, ringneck snake, king snake, thunder snake, vine snake
	elapid snake	sea snake, Indian cobra, green mamba
	viper	diamondback, horned viper, sidewinder
	boa snake	boa constrictor, rock python
salamander	newt	eft, common newt
	ambystomid salamander	spotted salamander, axolotl
insect	beetle	dung beetle, weevil, leaf beetle, tiger beetle, ladybug, rhinoceros beetle, long-horned beetle, ground beetle
	orthopterous insect	cricket, grasshopper
	dictyopterous insect	cockroach, mantis
	hymenopterous insect	bee, ant
	butterflyinsect	cabbage butterfly, lycaenid, monarch, admiral, sulphur butterfly, ringlet
	odonate insect	dragonfly, damselfly
	homopterous insect	cicada, leafhopper
furniture	table	desk, dining table
	baby bed	cradle, crib, bassinet
	seat	rocking chair, barber chair, park bench, throne, folding chair, toilet seat, studio couch
	lamp	table lamp
	cabinet	china cabinet, medicine chest
musical instrument	wind instrument	ocarina, flute, panpipe, oboe, cornet, sax, harmonica, bassoon, French horn, trombone
	stringed instrument	banjo, harp, violin, cello, acoustic guitar, electric guitar
	percussion instrument	steel drum, gong, marimba, drum, chime, maraca
	keyboard instrument	upright, grand piano, accordion, organ
scientific instrument	laboratory glassware	Petri dish
	magnifier	loupe, radio telescope
sports equipment	ball	golf ball, baseball, basketball, croquet ball
	gymnastic apparatus	parallel bars, balance beam, horizontal bar
	weight	barbell, dumbbell
electronic equipment	telephone	dial telephone, pay-phone, cellular telephone
	computer peripheral	printer, joystick, computer keyboard, mouse

	audio device	tape player, cassette player, CD player, iPod
	network device	modem
	display device	monitor, screen
clothing	bottoms (skirts)	hoopskirt, sarong, miniskirt, overskirt
	tops (sweaters)	sweatshirt, cardigan
	outwear	trench coat, poncho, fur coat
	swimwear	maillot, bikini, swimming trunks
	face & headwear	wig, sombrero, mortarboard, bonnet, mask, cowboy hat, bearskin
	nightwear	pajama
	protective wear	apron, knee pad, lab coat
	dresses & Gowns	gown
	underwear	brassiere
	footwear	sock, Christmas stocking
	neckwear	bow tie, bolo tie, Windsor tie
	traditional & formal Wear	abaya, kimono, vestment, academic gown
	wraps & shawls	stole, feather boa
container	reservoir	water tower, rain barrel
	bag	mailbag, plastic bag, backpack, purse
	jug	water jug, whiskey jug
	vessel	mortar, pitcher, tub, ladle, bucket, coffee mug
	bottle	wine bottle, beer bottle, pop bottle, water bottle, pill bottle
	basket	hamper, shopping basket
	box	mailbox, carton, pencil box, chest, crate
	glass	goblet, beer glass
	shaker	saltshaker, cocktail shaker
cooking utensil	pan	frying pan, wok
	cooker	Crock Pot
	pot	teapot, caldron, coffeepot
structure	monument	brass, megalith, triumphal arch, obelisk, totem pole
	religious building	church, mosque, boathouse, monastery, stupa
	housing	yurt, cliff dwelling, mobile home
	public building	planetarium, library
	movable structure	sliding door, turnstile
	supporting structure	plate rack, honeycomb, pedestal
	fence	stone wall, picket fence, chainlink fence, worm fence
	bridge	steel arch bridge, viaduct, suspension bridge
	residential structure	palace
	agricultural structure	greenhouse, barn, apiary

	commercial stucture	toyshop, restaurant, cinema, confectionery, bookshop, grocery store, tobacco shop, bakery, butcher shop, barbershop, shoe shop
	barrier	grille, bannister, breakwater, dam
	institutional structure	prison
tool	hand tool	hammer, plunger, screwdriver
	garden tool	lawn mower, shovel
	cutter	cleaver, plane, letter opener, hatchet
	power tool	chain saw
	opener	corkscrew, can opener
craft	sailing vessel	trimaran, schooner, catamaran
	boat	fireboat, canoe, yawl, gondola, speedboat, lifeboat
	ship	wreck, pirate, container ship, liner
	warship	aircraft carrier, submarine
	aircraft	airliner, warplane, airship, balloon
vehicle	bicycle	bicycle-built-for-two, mountain bike
	bus	minibus, school bus, trolleybus
	car	ambulance, beach wagon, cab, convertible, jeep, limousine, Model T, racer, sports car
	truck	fire engine, garbage truck, pickup, tow truck, trailer truck
	van	minivan, moving van, police van
	locomotive	electric locomotive, steam locomotive
	military vehicle	half track
	self-propelled vehicle	forklift, recreational vehicle, snowmobile, tank, tractor, golfcart, snowplow, go-kart, moped, streetcar, amphibious vehicle
	handcart	barrow, shopping cart
	sled	bobsled, dogsled
	train	bullet train
	wagon	horse cart, jinrikisha, oxcart
	wheeled vehicle	freight car, motor scooter, tricycle, unicycle
weapon	gun	rifle, assault rifle, revolver, cannon
	ranged weapon	missile, projectile

Table 3. **Complete hierarchy tree for our proposed ImageNet-3L dataset.**



(a) Original ImageNet's WordNet hierarchy

(b) Our 3-level hierarchy

Figure 10. **We restructure the original WordNet hierarchy into a clean, consistent three-level hierarchy for hierarchical recognition.**

C. Experiments on Messy Hierarchy

Setup. To evaluate robustness under the original WordNet structure, we construct a subset of ImageNet restricted to the *mammal* subtree. All ancestor nodes above *mammal* (e.g., *vertebrate*, *animal*, *entity*) are removed, and only classes under *mammal* are retained. This results in 120 fine-grained classes, with 152,548 training and 6,050 validation samples. The resulting hierarchy remains highly irregular (Fig. 11), with path lengths ranging from 5 to 10 levels. The distribution of chain lengths is skewed (e.g., most samples concentrate at depths 8–9, while deeper levels are sparse), leading to significant imbalance across levels.

Limitations of the Raw WordNet Hierarchy for Full-Path Prediction. Our goal follows the standard objective of hierarchical recognition [5, 7, 26], where a model predicts the full path across all levels. However, the raw WordNet hierarchy is not well-suited for this setting due to its irregular structure: classes have varying depths, making consistent level alignment impossible. To illustrate this, we construct a subset under the *mammal* hierarchy and perform full-path prediction on the raw structure (Fig. 11). We anchor all samples at the fine-grained level (Level 10) and propagate upward from *mammal*, leaving intermediate levels empty when necessary. While this enables leaf-level comparison, it introduces semantic misalignment across higher levels. For example, *hound* is compared with *sporting dog* rather than *spaniel*, and both *spaniel* and *English toy spaniel* appear at Level 8 despite their hierarchical inclusion. These issues lead to (1) *semantic inconsistency*, where nodes at the same level represent different levels of abstraction, and (2) *structural sparsity*, where certain levels contain very few classes (e.g., Level 5–7), resulting in weak supervision. While such hierarchies can still support flat classification with hierarchical penalties, they are **not suitable for full-path prediction**, which requires consistent semantic alignment across levels. This motivates the need to restructure and align the hierarchy, as in our ImageNet-3L, so that each level corresponds to a coherent semantic granularity.

Results and Analysis. Despite these challenges, Text-Attr (H-CAST) consistently improves performance over H-CAST across most levels (Level 3–Level 10), demonstrating robustness to structural noise. Levels 1 and 2 are excluded, as each contains only one class, resulting in trivial (100%) accuracy. The only exception is Level 9, where both methods perform poorly due to extreme sparsity and inconsistent supervision (some samples do not naturally have this level but are forced to predict it). These results highlight that the gains from *Text-Attr* are not tied to a well-structured hierarchy, but persist even in this ill-posed setting. At the same time, they motivate the need for a structured benchmark such as ImageNet-3L, which resolves these inconsistencies and provides a more meaningful evaluation of hierarchical recognition.

Table 4. **Performance under the original WordNet hierarchy**, which exhibits irregular depth and highly imbalanced levels across classes, making it less suitable for hierarchical evaluation. Despite this, we train and evaluate on the given structure as a robustness test, where Text-Attr provides effective semantic supervision and improves performance.

Depth (# class)	Level 10 (120)	Level 9 (3)	Level 8 (15)	Level 7 (8)	Level 6 (8)	Level 5 (4)	Level 4 (4)	Level 3 (2)
H-CAST	63.5	0.0	78.0	82.2	86.7	95.4	95.2	96.9
Text-Attr (H-CAST)	**67.7**	0.0	**80.0**	**83.8**	**87.5**	**96.6**	**96.0**	**98.0**

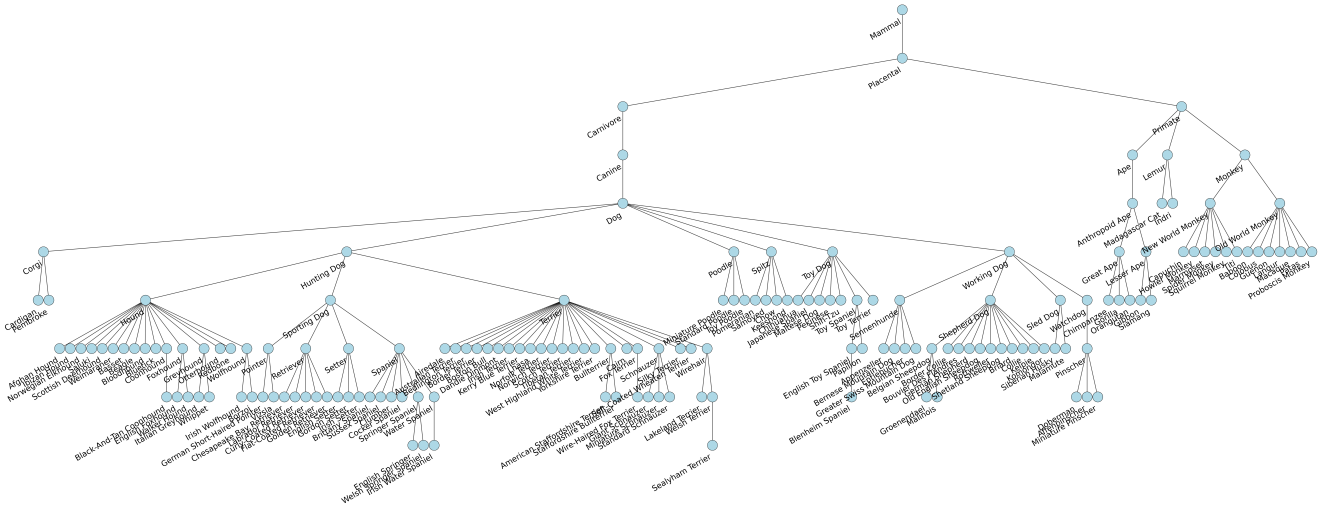


Figure 11. **Messy hierarchy of the ImageNet mammal subset.** The taxonomy exhibits irregular depths and inconsistent granularity across branches, making full-path prediction ill-posed. Despite these structural issues, our method consistently improves over the baseline, demonstrating robustness to noisy and misaligned hierarchies. Best viewed when zoomed in.

D. More Experimantal Results

D.1. Evaluaion on CUB-F

On the small-scale, single-domain dataset CUB-F (Table 5), Taxon-SSL achieves the best performance (63.96% FPA), showing the advantage of structured label propagation when per-class samples are scarce. Text-Attr methods perform moderately well (53.99–57.59% FPA) but are less effective here, as the bird-only domain limits textual diversity and reduces the benefit of language-based supervision. Still, they clearly outperform conventional hierarchical baselines (44.30% for HRN, 45.10% for H-CAST), underscoring the overall effectiveness of our approach. Unlike the trend on large-scale, diverse datasets such as ImageNet-F, where Text-Attr provides richer cues and stronger gains, these results confirm that there is no single recipe for free-grain learning: performance is tightly coupled with dataset characteristics, making the problem inherently challenging.

Table 5. **Taxon-SSL shows strong effectiveness on the small-scale dataset CUB-F, where label propagation provides reliable supervision.** Text-Attr methods are assumed to offer limited benefit due to the restricted textual diversity of this bird-only dataset.

CUB-F (13-38-200)	FPA (↑)	Species (↑)	family (↑)	Order (↑)	TICE (↓)
HRN [7]	44.30	46.72	81.20	96.36	27.15
H-CAST [26]	45.10	47.52	87.78	97.50	25.89
Taxon-SSL	**63.96**	**65.50**	**92.84**	<u>98.40</u>	**7.39**
Taxon-SSL + Text-Attr	<u>63.05</u>	<u>64.86</u>	<u>92.54</u>	98.38	<u>7.61</u>
Text-Attr (H-ViT)	57.59	59.10	91.60	98.05	10.72
Text-Attr (H-CAST)	53.99	55.58	91.72	**98.41**	18.95

D.2. Evaluation under Varying and Severe Label Sparsity Conditions

To evaluate model performance under diverse free-grain conditions, we experiment with various label availability ratios by randomly removing multi-level labels, e.g., (100%-60%-30%), (100%-50%-10%), and (100%-20%-10%), which represent the available proportions of basic, subordinate, and fine-grained labels, respectively. Each experiment is repeated with three different random seeds, and we report the average performance. The variance across runs was minor (0.1–1.8).

Consistent with our main results, these experiments (Table 6 & 7 & 8) also show that **there is *no* single method that performs best across all settings**. Instead, the most effective method varies depending on the dataset and the specific ratio of available labels, highlighting the importance of adaptable free-grain learning strategies.

For consistency, we refer to the three levels in CUB-Rand (order-family-species) and Aircraft-Rand (maker-family-model) as basic, subordinate, and fine-grained levels. We summarize the key findings below:

(1) Conventional hierarchical classification methods struggle under the free-grain setting, where label supervision is sparse and uneven across levels. For example, when labels are highly missing (e.g., only 10% available at the fine-grained level), HRN [7] and H-CAST [26] suffer more than a 50% drop in accuracy across all levels compared to the fully labeled (100%-100%-100%) setting on CUB-Rand (Fig. 6 & Table 8). This highlights the difficulty of the free-grain setting and the need for methods that can robustly handle incomplete supervision at multiple semantic levels.

(2) The performance of different methods varies with the amount of available supervision per class: Text-Attr methods perform better when more labeled samples are available, while Taxon-SSL is more effective under extreme label sparsity. For example, in Table 6, the average number of available fine-grained labels per class is approximately 9 for CUB-Rand and about 20 for Aircraft-Rand. Consistent with this difference, Taxon-SSL outperforms other methods on CUB-Rand, whereas Text-Attr (H-CAST) performs best on Aircraft-Rand. This trend persists across settings. In the most sparse setting, CUB-Rand (100-20-10, Table 8), where only about 3 fine-grained labels are available per class, Taxon-SSL shows a clear advantage. We attribute this to how supervision is utilized. Text-Attr relies on available labels and indirect semantic guidance via text features. In contrast, Taxon-SSL actively leverages unlabeled data through pseudo-labeling and strong augmentations, making it more effective when labeled examples are extremely limited.

(3) Sometimes, Taxon-SSL's high fine-grained accuracy comes at the cost of lower accuracy at higher levels in the taxonomy. For example, in Table 7, Taxon-SSL achieves the highest fine-grained accuracy (65.01%), but its subordinate and basic-level accuracies (85.53% and 92.81%) are lower than those of Text-Attr (H-CAST), which achieves 86.30% and 94.17%, respectively. This highlights a key challenge in free-grain learning: improving accuracy across all levels simultaneously is non-trivial, and optimizing for fine-grained performance alone may degrade consistency at coarser levels.

Table 6. **No single method performs best across all conditions—performance depends strongly on the amount of available supervision per class.** Text-Attr methods tend to perform better when more labeled samples are available, while Taxon-SSL is more effective under extreme label sparsity. For example, Taxon-SSL performs best on CUB-Rand with around 9 fine-grained labels per class, while Text-Attr (H-CAST) performs best on Aircraft-Rand with around 20, reflecting the impact of supervision density. These results highlight that method effectiveness is highly sensitive to label sparsity, emphasizing the need for adaptable approaches in free-grain learning.

Label Ratio	CUB-Rand (100%-60%-30%)					Aircraft-Rand (100%-60%-30%)				
	FPA(↑)	spec.(↑)	fam.(↑)	order(↑)	TICE(↓)	FPA(↑)	maker(↑)	fam.(↑)	model(↑)	TICE(↓)
HRN [7]	57.87	62.73	85.53	96.45	13.77	57.33	64.42	76.95	86.38	23.30
H-CAST [26]	61.88	67.36	90.05	94.32	13.04	64.67	68.88	85.58	91.43	13.76
Taxon-SSL	74.82	76.92	93.38	98.33	5.06	70.33	72.22	87.06	93.50	**7.18**
Taxon-SSL + Text-Attr	**74.90**	**76.95**	**93.41**	98.38	**4.91**	69.89	72.24	86.92	93.29	7.77
Text-Attr (H-ViT)	67.89	72.48	90.63	95.37	10.39	64.15	68.92	85.88	89.87	15.80
Text-Attr (H-CAST)	69.65	71.31	92.88	**98.48**	8.35	**71.43**	**73.56**	**89.66**	**95.31**	9.71

Table 7. **Maintaining accuracy across all hierarchy levels remains more challenging under sparse supervision.** For example, in 100%-50%-10% case, Taxon-SSL achieves the highest fine-grained accuracy (65.01%), but its subordinate and basic-level accuracies (85.53%, 92.81%) are lower than those of Text-Attr (H-CAST) (86.30%, 94.17%), which better preserves consistency across levels. This result illustrates the inherent difficulty of improving accuracy across all levels simultaneously, as objectives at different levels can be conflicting.

Label Ratio	Aircraft-Rand (100%-50%-10%)					Aircraft-Rand (100%-20%-10%)				
	FPA(↑)	maker(↑)	fam.(↑)	model(↑)	TICE(↓)	FPA(↑)	maker(↑)	fam.(↑)	model(↑)	TICE(↓)
HRN [7]	40.35	47.85	70.76	85.68	37.56	32.06	46.73	55.43	85.58	48.43
H-CAST [26]	47.57	51.93	78.31	87.11	28.42	40.33	45.44	67.28	84.12	35.61
Taxon-SSL	62.61	65.01	85.53	92.81	**10.22**	**58.73**	**61.10**	80.90	92.24	**11.77**
Taxon-SSL + Text-Attr	**62.95**	**65.49**	86.01	92.64	10.25	58.55	60.88	**80.97**	92.04	11.89
Text-Attr (H-ViT)	47.83	52.25	81.13	87.82	30.57	38.73	43.89	66.13	84.81	38.69
Text-Attr (H-CAST)	53.31	55.32	**86.30**	**94.17**	24.43	48.85	51.37	77.11	**93.01**	27.25

Table 8. **Taxon-SSL is more robust under extreme label sparsity.** In CUB-Rand (100%-20%-10%), where each class has only 3 fine-grained and 3 subordinate labels, Taxon-SSL achieves the best performance, while other methods struggle. HRN and H-CAST suffer over 50% drop in fine-grained accuracy compared to the fully-supervised (100%-100%-100%) setting. Text-Attr methods perform more robustly (10%+ higher than HRN/H-CAST), but still struggle under sparse supervision. We attribute this to how each method leverages supervision: Text-Attr depends on the provided labels and text-derived semantics, whereas Taxon-SSL can better exploit unlabeled data through pseudo-labeling and augmentations, leading to stronger performance when label sparsity is severe.

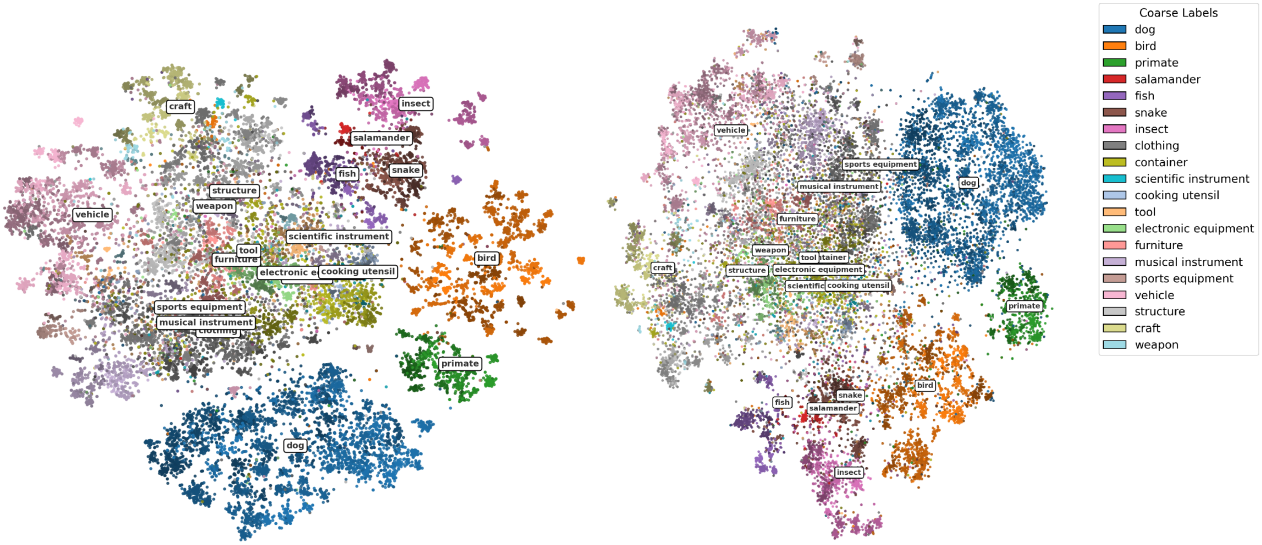
Label Ratio	CUB-Rand (100%-50%-10%)					CUB-Rand (100%-20%-10%)				
	FPA(↑)	spec.(↑)	fam.(↑)	order(↑)	TICE(↓)	FPA(↑)	spec.(↑)	fam.(↑)	order(↑)	TICE(↓)
HRN [7]	40.23	43.70	82.75	95.94	22.34	33.53	41.18	72.56	95.79	30.50
H-CAST [26]	39.03	43.41	85.74	93.23	24.60	32.97	38.66	76.89	92.50	29.43
Taxon-SSL	62.40	64.14	**92.33**	**98.26**	**6.01**	**59.18**	**61.44**	**89.79**	**98.20**	**7.65**
Taxon-SSL + Text-Attr	**62.52**	**64.87**	87.94	94.45	8.98	57.98	60.59	89.42	98.12	8.39
Text-Attr (H-ViT)	47.42	50.74	88.22	94.67	18.09	42.46	46.99	80.92	94.43	20.27
Text-Attr (H-CAST)	44.63	45.89	91.06	98.19	22.72	40.41	42.76	84.24	97.97	24.05

E. t-SNE Visualization

We visualize ImageNet-F embeddings of Text-Attr (H-CAST) and Taxon-SSL using t-SNE [21] to assess whether the learned representations capture semantic and hierarchical structure. Each point denotes an image embedding, colored by its basic-level class (20 categories), with brightness variations indicating fine-grained subclasses (505 total).

Both Text-Attr (H-CAST) and Taxon-SSL produce well-separated clusters consistent with the basic-level taxonomy, showing that coarse groupings are reliably captured. The key difference lies within coarse categories: **Text-Attr (H-CAST) reveals more distinct fine-grained subclusters** (e.g., breeds within *dog*, species within *bird*), whereas **Taxon-SSL yields tighter coarse clusters with less apparent fine-level separation**.

This contrast reflects their supervision signals. Text-Attr leverages diverse textual cues (attributes, parts, appearance terms), which promote discriminative, attribute-aligned features and sharpen within-class distinctions. Taxon-SSL, by propagating labels along the taxonomy and enforcing consistency under mixed-granularity supervision, regularizes embeddings within each coarse class and reduces intra-class variance—emphasizing coarse alignment over fine-level separability.



(a) Text-Attr (H-CAST) (b) Taxon-SSL

Figure 12. **t-sne Visualization on ImageNet-F.** Both methods separate coarse-level taxonomy well, but Text-Attr (H-CAST) yields clearer fine-grained subclusters (e.g., distinct groups within *dog* and *bird*) with more compact grouping, whereas Taxon-SSL shows some overlap of embeddings near cluster boundaries. This is likely due to ImageNet-F 's diverse large-scale categories, where text supervision provides rich attribute cues that sharpen fine-level distinctions.

F. Ablation Study

F.1. Importance of Text-guided Pseudo Attributes

Text-guided Pseudo Attributes jointly optimizes hierarchical label supervision ($\mathcal{L}_{\text{hier}}$) and text-guided pseudo attributes ($\mathcal{L}_{\text{text}}$) to learn semantically rich features: $\mathcal{L} = \mathcal{L}_{\text{hier}} + \alpha\mathcal{L}_{\text{text}}$ Fig. 13 quantifies $\mathcal{L}_{\text{text}}$'s impact by varying its weight α on CUB-Rand. Ablating $\mathcal{L}_{\text{text}}$ ($\alpha = 0$) causes a 5% absolute decline in both fine-grained accuracy and FPA compared to the optimal configuration ($\alpha = 0$). This gap underscores two key roles of text guidance: (1) it injects complementary visual semantics absent in class labels alone, and (2) it enforces attribute consistency across hierarchy levels. The performance recovery at ($\alpha = 1$) confirms that textual pseudo-attributes mitigate annotation sparsity while preserving taxonomic coherence.

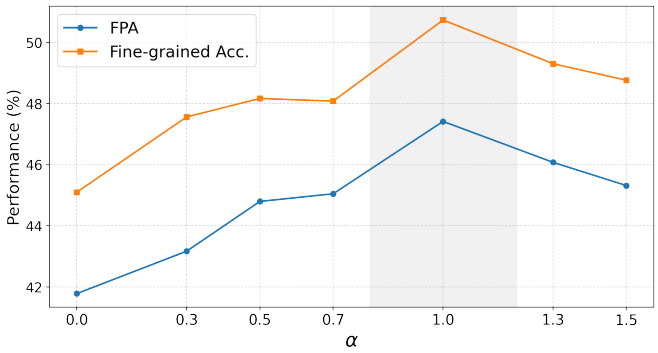


Figure 13. **Tuning α balances accuracy and taxonomic consistency.** At $\alpha = 1$ (optimal), Text-Attr (H-ViT) achieves peak fine-grained accuracy (blue) while maintaining hierarchical consistency (orange). Ablating $\mathcal{L}_{\text{text}}$ ($\alpha = 0$) causes a 5% accuracy drop and increased inconsistency, as class embeddings lose text-guided attribute alignment. Higher $\alpha > 1.0$ over-regularizes features, marginally degrading both metrics. This trade-off underscores the need to weight text supervision to resolve sparse annotations without distorting the hierarchy.

F.2. Combining Text-Attr and Taxon-SSL

We compare different training schedules for combining Text-Attr and Taxon-SSL on CUB-F. In the **joint setting**, both objectives are optimized simultaneously for 100 epochs. In the **two-stage setting**, we first train with one objective for 50 epochs and then add the other for the remaining 50 epochs, considering both orders: (1) Taxon-SSL $\rightarrow$ Text-Attr, and (2) Text-Attr $\rightarrow$ Taxon-SSL.

Table 9 show that starting with Text-Attr and then adding Taxon-SSL yields slightly higher full-path accuracy, likely because textual supervision promotes diverse feature learning before label propagation. In contrast, beginning with Taxon-SSL provides no advantage, and both two-stage variants perform similarly to joint training overall. Interestingly, joint training achieves higher consistency as measured by TICE. Given its simplicity and competitive performance, we adopt the joint strategy as our default.

Table 9. **Comparison of joint vs. two-stage training schedules for Text-Attr and Taxon-SSL on CUB-F.** While two-stage training (Text-Attr $\rightarrow$ Taxon-SSL) yields slightly higher accuracy, joint learning is simpler and provides better consistency (TICE).

CUB-F (13-38-200)	FPA (↑)	Species (↑)	family (↑)	Order (↑)	TICE (↓)
Taxon-SSL + Text-Attr (100 epochs)	<u>63.04</u>	<u>64.86</u>	<u>92.54</u>	**98.37**	**7.61**
Taxon-SSL (50 epochs) $\rightarrow$ +Text-Attr (50 epochs)	62.84	64.42	92.47	98.20	8.19
Text-Attr (50 epochs) $\rightarrow$ +Taxon-SSL (50 epochs)	**63.63**	**65.34**	**92.56**	<u>98.27</u>	<u>8.06</u>

F.3. Ablation on the Text Encoder in Text-Attr

Text-Attr relies on a text encoder to embed visual descriptions. To assess whether performance depends on the choice of text encoder, we replace CLIP with alternative encoders, including SigLIP [48] (*siglip-base-patch16-224*) and E5 [40] (*e5-base-v2*). All models use base configurations with a 768-dimensional embedding, and are integrated into our framework with minimal modification. As shown in Table 10, all encoders achieve comparable performance, with only minor differences across metrics. The text-only model (E5) performs slightly worse than CLIP and SigLIP, but the overall trend remains consistent. These results suggest that the gains of Text-Attr are not tied to a specific text encoder, but rather stem from the proposed training formulation for learning visual attributes.

Table 10. **Ablation on the text encoder used in Text-Attr on ImageNet-F**. Replacing CLIP with SigLIP and E5 results in comparable performance, with a slight drop for the text-only encoder (E5), indicating that gains mainly come from the proposed training formulation rather than the choice of encoder.

Text encoder	FPA (↑)	fine (↑)	sub. (↑)	basic (↑)	TICE (↓)
CLIP	**63.2**	**64.9**	**84.5**	93.6	**18.6**
SigLIP	62.9	**64.9**	84.3	**93.7**	19.5
E5	61.8	63.9	84.1	93.4	20.6

F.4. Ablation on Hierarchical Supervision in ViT

We further examine the architectural design choice of where to inject hierarchical supervision in the Vision Transformer (ViT) in Table 11. On CUB-F, we map the three taxonomy levels (Order–Family–Species) to different layers and compare multiple configurations: (6th, 9th, 12th), (8th, 10th, 12th), and (10th, 11th, 12th).

Among these, supervision at the 8th, 10th, and 12th layers yields the best performance. We interpret this as a balance between early and late representation learning: assigning hierarchy too early (e.g., 6–9–12) forces the model to align coarse categories before sufficient visual features are developed, while placing all supervision too late (e.g., 10–11–12) limits the model's capacity to gradually refine class granularity. The 8–10–12 configuration provides an appropriate middle ground, where lower-level categories benefit from moderately abstract features, and finer distinctions are introduced after the backbone has matured.

Table 11. **Performance comparison of different layer assignments for hierarchical supervision in ViT on CUB-F.** The 8th–10th–12th configuration achieves the best results, balancing early and late feature abstraction.

CUB-F (13-38-200)	FPA (↑)	Species (↑)	family (↑)	Order (↑)	TICE (↓)
6-9-12th layer	54.80	58.16	88.97	95.01	16.79
8-10-12th layer	**57.59**	**59.10**	**91.60**	**98.05**	**10.72**
10-11-12th layer	56.40	58.56	90.80	97.08	13.48

F.5. Ablation on Captioning Strategy and Caption Cost

Text-Attr relies on VLM-generated descriptions to capture visual attributes. To evaluate the benefit and cost of this step, we compare our approach with a simpler alternative that uses class-level text prompts (e.g., "a photo of [deepest available label]") instead of image-level descriptions. As shown in Table 12, image-level captions consistently outperform the simple text baseline across all metrics (+1.2 FPA), demonstrating the advantage of incorporating instance-specific visual descriptions. Nevertheless, the class-level text remains a strong baseline, indicating that even minimal textual supervision is effective.

Caption Cost. Generating image-level captions takes approximately 2.1 seconds per image on a single A40 GPU. This cost is incurred only once during preprocessing and can be further reduced through parallelization across multiple GPUs.

Table 12. **Comparison between image-level captions and simple class-level text**. Image-level descriptions improve performance across all metrics, while the simple text baseline remains competitive.

Caption	FPA (↑)	fine (↑)	sub. (↑)	basic (↑)	TICE (↓)
Simple Text	62.0	64.1	84.2	93.4	20.4
Ours (Image-level)	**63.2**	**64.9**	**84.5**	**93.6**	**18.6**

G. Implementation Details

For ViT [8] models, we use ViT-Small for Text-Attr (H-ViT) and Taxon-SSL and H-CAST-Small [26] for Text-Attr (H-CAST) to match parameter sizes.

For Text-Attr (H-ViT), we insert fully-connected layers to the class token at the 8th, 10th, and 12th layers for basic, subordinate, and fine-grained supervision. The 12th-layer patch features are projected to match the text embedding dimension via an FC layer. For Text-Attr (H-CAST), hierarchical supervision is applied to the last three blocks, following [26]. Due to low dimensionality in the final block, we align text features with the features of the second block. For Text-Attr methods, CLIP-ViT-B/32 is used to extract text embeddings, which remain frozen during training.

In Taxon-SSL, we apply a shared MLP to the class token from the final (12th) layer, followed by three separate linear classifiers for basic, subordinate, and fine-grained supervision. When combined with Text-Attr, we additionally project the class token through a linear layer and align it with the corresponding text feature.

For hierarchical classification baselines, HRN [7] and H-CAST [26], we follow their original training protocols and retrain them under our free-grain setting. We extend HRN to handle missing labels at two levels instead of one. For H-CAST, we provide supervision using the available labels at each corresponding level. Full hyperparameter configurations are provided in Table 13.

We train all models for 100 epochs, except for ImageNet-F, which are trained for 200 epochs due to the larger scale. All experiments were conducted on an NVIDIA A40 GPU with 48GB memory. We used a single GPU for all experiments, except for ImageNet-F, which was trained using 4 GPUs.

Table 13. **Hyperparameters for training Text-Attr (H-ViT), Text-Attr (H-CAST), and Taxon-SSL.** We follow the training setup of H-CAST [26] for Text-Attr methods (Text-Attr (H-ViT) and Text-Attr (H-CAST)), and adopt the settings of CHMatch [44] for Taxon-SSL.

Parameter	Text-Attr (H-ViT)	Text-Attr (H-CAST)	Taxon-SSL
batch_size	256	256	128
crop_size	224	224	224
learning_rate	$5e-4$	$5e-4$	$1e-3$
weight_decay	0.05	0.05	0.05
momentum	0.9	0.9	0.9
warmup_epochs	5	5	0
warmup_learning_rate	$1e{-}6$	$1e{-}6$	N/A
optimizer	Adam	Adam	SGD
learning_rate_policy	Cosine decay	Cosine decay	Cosine decay
α (weight for $\mathcal{L}_{\text{text}}$)	1	1	1 (for +Text-Attr)

H. Full Losses for Taxonomy-guided Semi-Supervised Learning (Taxon-SSL)

In this we provide full details of Taxonomy-guided Semi-Supervised Learning (Taxon-SSL). As described in Sec. 4.3, following standard practice [44], our classifier consists of a shared feature extractor f_{feat} and level-specific heads $\{h_l\}_{l \in \mathcal{S}_x}$. For supervised samples with known labels $y_1, \ldots, y_L$ across L taxonomy levels, we apply a per-level hierarchical supervision loss $\mathcal{L}_{\text{sup}}$:

$$\mathcal{L}_{\text{sup}} = \sum_{l=1}^{L} \mathbb{1}_{\{y_l \text{ exists}\}} \cdot \mathcal{L}(h_l(f(x)), y_l). \tag{5}$$

For unlabeled levels, we generate a weakly augmented image ($\mathcal{W}(x)$) and a strongly augmented version ($\mathcal{S}(x)$). Confident predictions from $\mathcal{W}(x)$ at each levels become pseudo-labels for $\mathcal{S}(x)$, denoted as $\mathcal{PL}_l(x)$. Concretely, our pseudo-labeling loss $\mathcal{L}_{\text{pl}}$ is

$$\mathcal{L}_{\text{pl}} = \sum_{l=1}^{L} \mathbb{1}_{\{\mathcal{PL}_l(x) \text{ is over confidence threshold}\}} \cdot \mathcal{L}(h_l(f(\mathcal{S}(x))), \mathcal{PL}_l(x)). \tag{6}$$

Confidence thresholds follow the percentile-based schedule introduced in CHMatch [44].

Lastly, to reinforce hierarchical structure in the learned representation, we construct a taxonomy-aligned affinity graph. Within each mini-batch $\mathcal{B}$, we construct affinity graphs W^l for each hierarchy, where $W^l_{ij} = 1$ if the ith and jth image have the same pseudo-labels and $W^l_{ij} = 0$ otherwise. Then the taxonomy-aligned affinity graph W is defined by

$$W_{ij} = \begin{cases} 1 & \text{if } W^1_{ij} = \ldots = W^L_{ij} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

We pull together positive pairs where $W_{ij} = 1$ and push apart negative pairs where $W_{ij} = 0$. Formally, $\mathcal{L}_{\text{tacl}}$ is defined by

$$\mathcal{L}_{\text{tacl}} = -\frac{1}{\sum_j W_{ij}} \cdot \log \frac{\sum_j W_{ij} \exp((g(f(x_i)) \cdot g(f(x_j))')/t)}{\sum_j (1 - W_{ij}) \exp((g(f(x_i)) \cdot g(f(x_j))')/t)}, \tag{8}$$

where i is the index of current image, g is the projection head for $\mathcal{L}_{\text{sup}}$, and t is the temperature hyperparameter.

Our final training objective is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{hier}} + \beta \cdot \mathcal{L}_{\text{pl}} + \gamma \cdot \mathcal{L}_{\text{tacl}}, \tag{9}$$

where β and γ control the relative contributions of the loss terms, and we set both to 1 in all experiments.

I. Related Work

Hierarchical recognition has been used to refer to different tasks. A large body of work leverages taxonomies only to improve **leaf-node prediction** [12, 17, 47, 49]. Thus, evaluation in these works typically relies on top-1 accuracy or mistake severity at the leaf level. These methods predict a *single fine-grained label* and assume the full hierarchy can be recovered from it. However, fine-grained prediction is often impossible in real-world scenarios due to visual ambiguity, leading these models to produce uninformative outputs.

In this paper, we focus on **full-taxonomy prediction** [5, 16, 26, 41], where the model must output labels at all levels of the taxonomy. This setting introduces cross-level inconsistency, as predictions across levels must align with the hierarchical structure. Recent work [36] shows that such inconsistencies arise even in large vision–language models like GPT-4o [15] and Qwen2.5-VL-72B [3], underscoring the difficulty of the problem.

Existing full-taxonomy prediction methods enforce consistency constraints and demonstrate strong performance [5, 16, 26, 41]. However, they assume that all hierarchical levels are fully annotated for every training sample, which is unrealistic. In practice, annotation granularity naturally varies due to visual ambiguity or annotator expertise—some images may only receive a coarse label (e.g., *bird*), while others have fine-grained labels (e.g., *bank swallow*).

To address this gap, we study a more realistic setting, free-grain learning, where supervision may appear at any level of the taxonomy and the model must infer the complete label path from partially observed labels. Existing approaches related to partial hierarchical supervision do not fully capture this setting. HRN [7] considers partial labels but only in a limited two-level scenario created by randomly removing fine-grained labels, which does not reflect the structured ambiguity found in real taxonomies. Kim et al. [18] also incorporates mixed-granularity labels, but treats them in a flat manner, overlooking the hierarchical relationships essential to our formulation. These studies were also restricted to small datasets such as CUB [43], whereas our work establishes a general free-grain formulation and provides a large-scale benchmark to study it systematically.

Hierarchical recognition datasets. Full-taxonomy prediction requires datasets where each hierarchical level is meaningful to predict. However, large-scale datasets like ImageNet [32] are not designed for this purpose. Its WordNet [10] taxonomy has uneven depths (e.g., *minivan* appears around the 15th level while *teddy bear* appears around the 7th), and contains many abstract nodes such as *entity*, *object*, or *whole* that are **not useful prediction targets** (Fig. 3). Such a hierarchy supports only leaf-node prediction with mistake-severity penalties [12], where the model still predicts a single leaf label and the hierarchy is used merely to score how far an error is from the ground truth.

Because of this limitation, prior full-taxonomy prediction work has relied on small, clean datasets like CUB [43] and Aircraft [22], where the hierarchy is well-defined but the scale is limited. iNaturalist [39] provides a deeper taxonomy, but its scope is restricted to biological species and does not generalize to broad visual domains.

To enable large-scale and general hierarchical recognition, we introduce ImageNet-3L, which provides three semantically meaningful levels without the abstract superordinate nodes (e.g., *entity*, *object*) present in WordNet. We center the hierarchy on Rosch's basic-level categories [31], the level at which humans naturally identify objects (e.g., *bird*, *vehicle*), and organize categories downward into subordinate and fine-grained levels. This produces a clean and meaningful three-level taxonomy that focuses on distinctions worth predicting and is well suited for full-taxonomy recognition. Using this, we create free-grain variants to study hierarchical prediction under varying label granularity.

Long-tailed recognition has been extensively studied [2, 14, 20, 24, 25, 29, 38, 42, 50], mostly focusing on imbalance at a single fine-grained level (*inter-class* imbalance). In contrast, we address both *inter-class imbalance* (across classes) and *intra-hierarchy imbalance* (across semantic levels) in a hierarchical setting, where classes themselves may be balanced but label granularity varies across them. DeepRTC [45] considers taxonomies, but aims to improve inference reliability via early stopping rather than predicting the full taxonomy.

Semi-supervised learning typically combines labeled and unlabeled data at a single fine-grained level [4, 34, 37]. Recent work incorporates coarse labels [11, 44], but still targets fine-grained accuracy. In contrast, our setting demands consistent prediction across the full taxonomy with heterogeneous supervision, making existing methods not directly applicable.

Weakly-supervised recognition typically aims to predict fine-grained labels when only coarse labels are available during training [13, 30]. These methods assume fully observed labels at a coarse level and focus on improving predictions at a fine-grained level. In contrast, our setting requires handling multi-granularity labels and inferring the full taxonomy.

Large-language models for recognition. Recent approaches [19, 27, 33, 51] leverage vision–language models (VLMs) (e.g., CLIP [28]) or large language models (LLMs) (e.g., GPT-4 [1]) by generating textual descriptions from label names and feeding them into an LLM to improve flat classification. Their primary goal is to perform label-driven reasoning without training a new visual model. In contrast, we do not use labels to expand label descriptions. Instead, we use VLMs to extract

textual cues directly from the image, without referencing labels, so that the image encoder can learn visual attributes shared across hierarchical levels when supervision is incomplete. At inference, our model is image-only and does not rely on VLMs or LLMs, since our goal is hierarchical prediction rather than label-driven reasoning.