A. Appendix A

A.1. Experimental Details

A.1.1. Baselines

As fine-grained visual recognition without expert annotations is an emerging area, there are limited existing baselines, with FineR [18] and CLEVER [3] being notable exceptions. To provide a comprehensive comparison, we also include below-mentioned strong baselines. (i) CLIP Zero-Shot Upper Bound (UB), which uses the ground-truth class names as text prompts, reflecting expert-level knowledge and serving as an upper performance bound. (ii) WordNet Baseline, which uses CLIP with a large vocabulary of 119,000 nouns from WordNet [20]. (iii) BLIP-2 [15] and Flan-T5xxl [14], a VQA-based approach that identifies the main object in an image via the prompt "What is the name of the main object in this image?". (iv) SCD [7], which first clusters images and then narrows down labels using CLIP with a combined vocabulary from WordNet and Wikipedia bird names. (v) CaSED [5], which retrieves captions from a large-scale knowledge base and extracts class names by parsing and classifying nouns with CLIP. (vi) KMeans clustering on CLIP visual features. (vii) Sinkhorn-Knopp Clustering, a parametric method, applied with features from CLIP and DINO. All baselines are evaluated using the CLIP ViT-B/16 vision encoder.

A.1.2. Implementation Details

For the Class Names Reasoning and Class Names Refinement modules, following [18], BLIP-2 [15] with Flan-T5xxl [4] are used as the visual question answering (VQA) model, ChatGPT (gpt-3.5-turbo) accessed via the OpenAI API as the large language model (LLM), and CLIP ViT-B/16 as the vision-language model (VLM). The hyperparameters for multi-modal fusion and data augmentation are set to $\alpha = 0.7$ and $K = 10$, respectively.

For the Contextual Grounding module, we use Google Gemini 2.0 LLM accessed via the public API. Where we prompt the LLM to generate 100 in-context sentences about the guessed class name. These 100 sentences per class are then averaged and normalized. Specifics about the prompt and other details will be publicly available in our shared repository. In the Class Names Refinement module, a larger CLIP model based on ViT-L/14 is used for filtration, and finally, in the Vision-Language Coupling block we utilize the smaller ViT-B/16 base model for faster inference.

A.1.3. Prompt Design

The exact prompt used with Gemini-2.0 large language model to obtain in-context class-specific sentences (specifically applied for the CUB-200 dataset, "classname" word will be replaced with an actual guessed label from the classname reasoning step):

```
Generate 100 short and common sentences with noun
{classname}, a type of bird, as a main subject.

This noun should only be used in a realistic
and descriptive general context with various real
and related scenarios. In the sentence, highlight
something specific about the classname, a type
of bird, which helps to distinct it from other
birds (it can be its color, shape, size,
background, and so on).

Only use the main and original sense of this noun,
no idioms. Only use visually descriptive adjectives
or participles. Each sentence should be between
5 to 8 words (excluding the noun). Do not use
the possessive form. Do not add an article at
the beginning of the sentence. Do not repeat
the noun in the same sentence. Do not capitalize
the first letter of the sentence unless this is
a name. Do not add a dot at the end of sentence.
Make sure sentences are diverse and do not repeat
each other.

Make sure the noun is included in each sentence.
Make sure the sentences are between 5 to 8 words each.

Return output in the following structure as a single
line: ["<generated_sentence_1>",
"<generated_sentence_2>", ...,
"<generated_sentence_n>"]
```

A.2. Additional Analysis

A.2.1. Ablation study

In this section we assess the contribution of our key design components: Class-specific Contextual Grounding (CCG) and Class Names Refinement (CNR). For this, we perform an ablation study on the Stanford Dogs dataset, with results shown in Table 4. The results demonstrate that both components contribute meaningfully to performance. Adding CCG alone already improves clustering accuracy (cACC) and semantic accuracy (sACC) compared to the baseline (no CCG or CNR), increasing cACC from 51.30% to 51.86%, and sACC from 65.41% to 66.98%. This suggests that incorporating context tailored to each class helps align the discovered clusters more effectively.

Components		**Accuracy**		**Sensitivity**	
CCG	CNR	cACC ↑	sACC ↑	FN ↓	TP ↑
×	×	51.30	65.41	7	52
✓	×	51.86	66.98	4	55
✓	✓	**51.99**	**67.11**	**0**	**59**

Table 4. Ablation study for our proposed components. The performance is reported for the Stanford Dogs dataset for a fixed run. Acronyms are: Class-specific Contextual Grounding (CCG), Class Names Refinement (CNR), Clustering accuracy (cACC), Semantic accuracy (sACC). The number of filtered (unused) real class names is denoted as False Negative (FN), and the number of kept (used) real class names as True Positive (TP). Best results are in bold.

For the sensitivity analysis, we compare each guessed class name with the ground truth labels. If a guessed name fully matches any of the actual labels, then it is chosen for analysis and disregarded otherwise. Next, the class name is considered as True Positive if it was correctly guessed and used further for the classification, and as False Negative if it was correctly guessed but was filtered out at the Class Names Refinement stage (and was not used for the classification). It can be observed that when both CCG and CNR are enabled, the system achieves the highest accuracy and sensitivity, with cACC of 51.99% and sACC of 67.11%. Importantly, the final configuration results in zero false negatives (FN = 0) with no real class names mistakenly filtered out while retaining all 59 ground-truth class names (TP = 59). This highlights the ability of our refinement mechanism to retain all semantically relevant classes.

In summary, both CCG and CNR contribute complementary benefits: CCG enriches semantic grounding, while CNR ensures high recall in class selection. Their combination is critical for robust and precise vocabulary-free classification in fine-grained domains.