Waffling around for Performance: Visual Classification with Random Words and Broad Concepts

Karsten Roth[1,*], Jae Myung Kim[1,*], A. Sophia Koepke[1], Oriol Vinyals[2], Cordelia Schmid[3], Zeynep Akata[1,4]

[1]University of Tübingen, Tübingen AI Center, [2]Google DeepMind,

[3]Inria, Ecole normale supérieure, CNRS, PSL Research University, [4]MPI for Intelligent Systems

[*]equal contribution

Abstract

The visual classification performance of vision-language models such as CLIP has been shown to benefit from additional semantic knowledge from large language models (LLMs) such as GPT-3. In particular, averaging over LLM-generated class descriptors, e.g. "waffle, which has a round shape", can notably improve generalization performance. In this work, we critically study this behavior and propose `WaffleCLIP`, a framework for zero-shot visual classification which simply replaces LLM-generated descriptors with random character and word descriptors. ***Without*** *querying external models, we achieve comparable performance gains on a large number of visual classification tasks. This allows `WaffleCLIP` to both serve as a low-cost alternative, as well as a sanity check for any future LLM-based vision-language model extensions. We conduct an extensive experimental study on the impact and shortcomings of additional semantics introduced with LLM-generated descriptors, and showcase how - if available - semantic context is better leveraged by querying LLMs for high-level concepts, which we show can be done to jointly resolve potential class name ambiguities. Code is available here: https://github.com/ExplainableML/WaffleCLIP.*

1. Introduction

Task-specific tuning of natural language prompts [31, 67, 8, 26] can improve the performance of large vision-language models (VLMs) [47]. However, if the model does not have access to additional training data, i.e. in the zero-shot setting, this is not an option. Instead, a promising alternative [42, 46, 36] is querying large language models (LLMs) to provide additional semantic context to enrich class representations. Extending classnames with fine-grained class descriptors generated by GPT-3 [5] and minimal human intervention has experimentally shown to boost results [36, 46], for instance with class-based descriptors on top of classnames, e.g. *a round shape* for *waffle* [36].

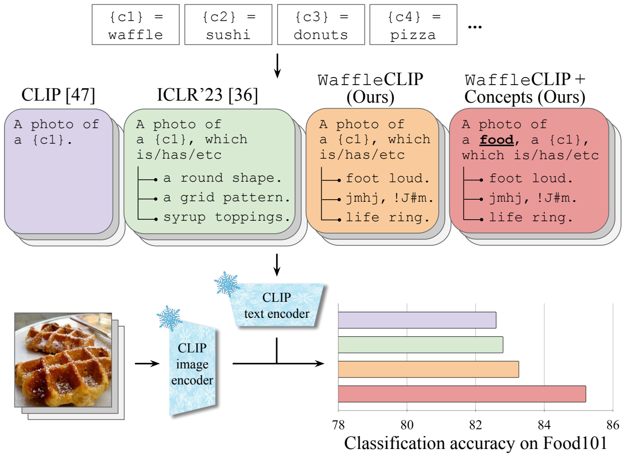

Figure 1: Substituting GPT-3 generated fine-grained descriptors with random word or character sequences yields competitive performance. High-level concepts further resolve classname ambiguities for additional gains.

However, close inspection of GPT-3 generated semantic cues indicates a high degree of diversity, limited visual relevance, and ambiguity [36]. For instance, multiple descriptors can be assigned to the same class despite likely not co-occurring, e.g. *"steamed"* and *"fried"*, or non-visual attributes might be mentioned, e.g. *"a sour and spicy smell"*, or the class interpretation might be ambiguous, e.g. *"webbed feet"* for *"Peking duck"* as a food item. Hence, the underlying drivers of performance improvements when using generated fine-grained class descriptors are unclear.

To understand these performance gains, we first show that each set of class-specific GPT-3 generated descriptors can be replaced with a fixed set of randomly selected, class-independent descriptors while still retaining similar benefits in performance. Motivated by this observation, we take this one step further and propose `WaffleCLIP`, named after *waffling around* the class name, that replaces the LLM-generated fine-grained descriptors, e.g. *a round shape, a grid pattern*, with random words (e.g. *"foot loud"*) or character lists (e.g. *"jmhj, !J#m"*) based on average class name length and word counts (cf. Figure 1). As `WaffleCLIP` does not require access to LLMs for additional context (unlike e.g. [36, 42, 46, 56]), it remains *inherently zero-shot*. Consequently, it also serves as an important sanity check for future methods utilizing external model queries.

Naturally, the convincing performance of `WaffleCLIP` across benchmarks raises questions regarding the true benefits of additional semantics introduced by LLM-generated descriptors. We provide answers with extensive experiments, showcasing that semantic descriptors produced by LLMs offer a *structurally* different and *complementary* impact on the classification behavior. However, we find this not to be fully driven by additionally introduced semantics, but rather a different form of structured noise ensembling.

Instead, we show that actual semantic context is better introduced through coarse-grained, high-level concepts. Given access to external LLMs, we suggest a query mechanism for GPT-3 to automatically generate these concepts (e.g. *food* for *waffle, peking duck*), while jointly resolving issues of context-dependent class label ambiguity.

In summary, our contributions are: **1)** We motivate and propose `WaffleCLIP` to use random character and word descriptors to enhance the semantic retrieval process in VLMs (particularly CLIP); **2)** we demonstrate that `WaffleCLIP` yields comparable zero-shot classification performances at lower cost compared to methods reliant on LLM-generated descriptors, thus also serving as an important sanity check for future models; **3)** we extensively study the semantic context introduced through LLM-generated descriptors and propose (automatically extracted) high-level LLM-generated concepts as an alternative for better use of semantics while tackling classname ambiguities.

2. Related Work

Image classification with VLMs such as CLIP [47] has gained popularity particularly in low-data regimes. As input prompts have a significant impact on the performance, recent research has focused on the exploration of learnable prompts for the text encoder [67, 66, 33, 54], the visual encoder [1, 7, 61, 32] or for both encoders jointly [62].

Alternatively, synthetic images generated using available classnames can support corresponding image classification [56, 2, 20]. In contrast, we do not tune prompts or query external image generation methods, but propose to use prompts containing random characters or words to enhance the zero-shot capabilities of VLMs.

Adding external knowledge to language prompts. Recently, multiple works have shown how LLMs can be leveraged to obtain more effective prompts. [46, 38, 35] utilized GPT-3 [5] to produce and study lengthy, descriptive sentences that articulate the visual concepts for each category, while [42] generated semantic hierarchies to identify subclasses of categories for zero-shot class prediction. [36] used multiple fine-grained LLM-generated class descriptors, which enhance accuracy and appear to provide interpretability by assigning weights to each descriptor.

Similarly, different kinds of descriptions have been used for image classification tasks, by manually crafting descriptions [48, 21], or by utilizing external databases based on e.g. Wikipedia [16, 45, 39, 11], the WordNet hierarchy [37, 52, 49], or the ImageNet-Wiki [6].

Whilst external knowledge from LLMs can be valuable, in this work we show that one can match the image classification performance gains of using fine-grained LLM-generated descriptors with randomly sampled characters and words as class descriptors. In addition, we find that if semantic context is available through LLMs, it is better integrated through high-level context (c.f. also [15]), for which we provide an automatic extraction mechanism.

Noise augmentation. Data augmentation through noise is known to enhance the performance and robustness of model training for a variety of tasks and domains [53, 17]. In the language domain, noise can be incorporated in the embedding or input space. For instance, [55, 9, 19] used linguistic embedding space augmentations inspired by mixup [64], and [10] added Gaussian embedding space noise. Augmentation through input space noise has been performed at the word- [28, 60], token- [60] or character-level [51, 22, 3, 41].

For character-level noise augmentation, characters are randomly substituted, added, or removed [22, 3, 41]. In all cases, these augmentation are used to prevent overfitting *during training*. Instead, our approach utilizes character- and word-level language augmentation to perturb the class prompts for improved zero-shot image classification.

3. Method

We first describe image classification using class descriptors following [36] (§3.1), before motivating and explaining our LLM-free, random semantic descriptor alternative `WaffleCLIP` (§3.2). Finally, if LLMs are available, we highlight a simple extension to incorporate semantics while jointly resolving ambiguities with automatically extracted high-level semantic concepts (§3.3).

3.1. Image classification with class descriptors

Given target categories C and a query image x, the zero-shot image classification protocol used in CLIP [47] defines the classification problem as nearest neighbour retrieval:

$$\tilde{c} = \arg\max_{c\in C} s(\phi_I(x), \phi_L(f(c))), \quad (1)$$

with prompt $f(c)$ = `"A photo of a {c}."` and image and language encoder ϕ_I and ϕ_L. To improve

ViT-B/32	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
CLIP [47]	54.71	62.01	51.28	40.78	39.12	82.59	85.06	43.18	57.34
DCLIP [36]	**55.82**	**63.12**	52.47	**43.29**	40.47	82.79	86.54	**43.99**	**58.56**
DCLIP (same, 1x)	55.47 ±0.24	62.89 ±0.19	52.64 ±0.28	39.74 ±2.69	40.29 ±0.47	83.82 ±0.48	87.04 ±0.27	43.35 ±0.41	58.16 ±1.01
DCLIP (same, 2x)	55.75 ±0.21	63.10 ±0.19	**52.72** ±0.23	39.73 ±1.66	**40.61** ±0.22	**84.01** ±0.23	**87.10** ±0.14	43.29 ±0.22	58.29 ±0.62

Table 1: **Motivating random class descriptors.** Comparing CLIP [47] and the GPT-descriptor-extended CLIP [36] (DCLIP) with the same set of randomly sampled descriptors for each class, where the set size is either the average number of descriptors per class in DCLIP (*same, 1x*), or twice that (*same, 2x*). A random set of descriptors per class can match or even outperform DCLIP across backbone architectures (results for ViT-L/14 and ResNet50 are included in the suppl. material) confirming that randomized prompt averaging leads to higher performance.

the retrieval process, [36] converts the simple class-embedding retrieval to a dictionary-based one, where a class c is associated with a set of descriptors D_c: "{c} which (is/has/etc) {descriptor}." with e.g. c = "waffle" and descriptor = "a round shape". Given D_c for classes c, classification is reformulated as

$$\arg\max_{c \in C} \frac{1}{|D_c|} \sum_{d \in D_c} s(\phi_I(x), \phi_L(d)), \tag{2}$$

which defines the similarity between the image x and class c as the average similarity to all its descriptor variants. We abbreviate this descriptor-based extension of CLIP as *DCLIP*.

3.2. WaffleCLIP

DCLIP [36] [1] requires external LLMs for descriptors that convert the single-class matching problem to one over an ensemble of fine-grained class representations.

3.2.1 Motivation

We observe that LLM-generated class descriptors reveal high diversity, limited visual relevance, and ambiguity. From a conceptual perspective, this makes it hard to pin down the precise benefits of generated class descriptors used e.g. in [46] or [36]. To understand a possible driver of performance improvements, we conduct a simple experimental study, shown in Tab. 1. We take all available LLM-generated descriptors for a dataset from [36], sample a small set of descriptors where the cardinality of the set is the average number of descriptors per class used in DCLIP, and assign this same set of random descriptors to every class, i.e. *DCLIP (same, 1x)*.

This shows a close match to DCLIP (e.g. 58.56% and 58.16% for ViT-B/32 in total average) and in parts even better performance (e.g. 0.83% improvement in Food101 for ViT-B/32). This reveals averaging over descriptor variations as one of the key drivers for performance. The results further improve when increasing the number of random LLM-generated descriptors for each class (*DCLIP (same, 2x)*, e.g. 58.16% → 58.29% for ViT-B/32).

Correspondingly, these results indicate that the role of additional descriptor semantics is likely overestimated, especially when uncurated descriptors are used. Building on the benefits of averaging over various prompt variants to extract a better semantic representation estimate of an associated class, we investigate whether fully randomized prompt descriptors can provide similar benefits, **without** querying external LLMs.

3.2.2 WaffleCLIP Method

This motivates WaffleCLIP, an *LLM-free* descriptor alternative that uses simple randomized descriptors. In particular, we populate D_c with class-independent, random word sequences or random character lists, with a fixed number of characters per word l_w, and a fixed number of words n_w. For example, $l_w = 4$ and $n_w = 2$ for char_seq_1 = "aks@, pg2f" in Fig. 2. To avoid introducing hyperparameters, we leverage a simple heuristic where the average number of words and average number of characters per word in the provided class labels determines l_w and n_w. As a result, this converts the standard CLIP input prompt "A photo of a {c}." into "A photo of a {c}, which (is/has/etc) {random_sequence}.", where we follow the extension structure used in [36].

3.3. Better semantics and reduced ambiguity via high-level concepts

Due to the limited impact of additional semantics introduced by fine-grained descriptors (c.f. §3.2), we propose an alternative way of querying LLMs that does not require averaging across multiple descriptors and simultaneously addresses the issue of class ambiguities. Therefore, we suggest taking a step back and searching not for additional class details, but instead for higher-level commonalities *between*

[1] DCLIP [36] reports improvements over CLIP by using the phrase "{c}, which (is/has/etc) {descriptor}." instead of "A photo of {c}, which (is/has/etc) {descriptor}." as suggested and studied in the original CLIP paper. For fair comparison with existing baseline CLIP results, we thus utilize the latter, but find similar behaviour for other prompt structures.

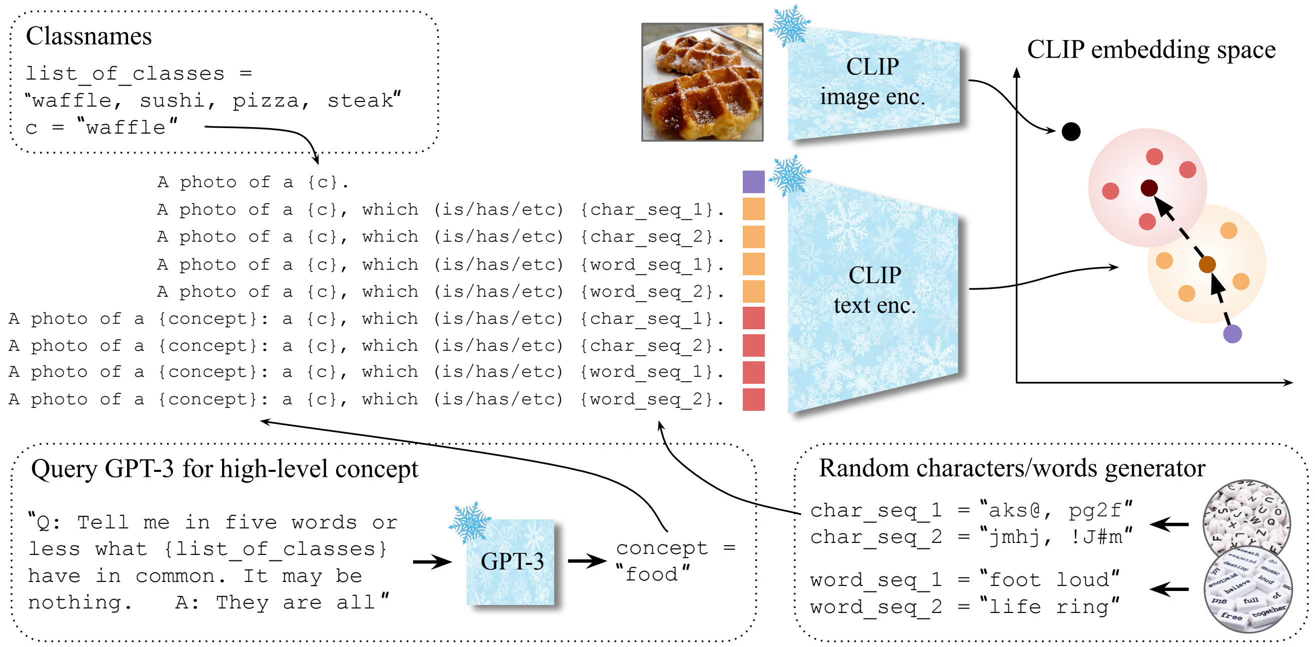
Classnames
list_of_classes =
"waffle, sushi, pizza, steak"
c = "waffle"
A photo of a {c}.
A photo of a {c}, which (is/has/etc) {char_seq_1}.
A photo of a {c}, which (is/has/etc) {char_seq_2}.
A photo of a {c}, which (is/has/etc) {word_seq_1}.
A photo of a {c}, which (is/has/etc) {word_seq_2}.
A photo of a {concept}: a {c}, which (is/has/etc) {char_seq_1}.
A photo of a {concept}: a {c}, which (is/has/etc) {char_seq_2}.
A photo of a {concept}: a {c}, which (is/has/etc) {word_seq_1}.
A photo of a {concept}: a {c}, which (is/has/etc) {word_seq_2}.
CLIP
image enc.
CLIP
text enc.
CLIP embedding space
Query GPT-3 for high-level concept
"Q: Tell me in five words or less what {list_of_classes} have in common. It may be nothing. A: They are all"
GPT-3
concept =
"food"
Random characters/words generator
char_seq_1 = "aks@, pg2f"
char_seq_2 = "jmhj, !J#m"
word_seq_1 = "foot loud"
word_seq_2 = "life ring"

Figure 2: **Visual classification with WaffleCLIP using random characters/words.** WaffleCLIP utilizes a collection of prompt variations by simply injecting character-level and word-level noise around the classname (*top left*, *orange*). Simple averaging consequently raises the robustness of the extracted semantics and the corresponding retrieval process (*top right*), leading to notable gains in the open-vocabulary classification performance using Vision-Language models such as CLIP, while approximating the performance improvements gained by e.g. querying external Large Language models for additional semantic descriptors. In addition, we show that if given access to external LLMs, WaffleCLIP can be further enhanced by adding a high-level concept descriptor in the prompt (*red*).

the classes, akin to the use of class hierarchies in image classification [18]. Understanding commonalities between multiple target classes can help resolve ambiguities. If the class "boxer" is seen in the context of animal classification, it likely refers to the animal instead of a human athlete.

We propose to automatically produce such high-level concepts by using available class names (or subsets if the class count exceeds the maximum LLM input sequence length) $C_\mathcal{D}$ for a dataset $\mathcal{D}$ and querying GPT-3 [5] with:

"Q: Tell me in five words or less what {list_of_classes} have in common. It may be nothing. A: They are all ".

or small variants thereof. After extracting the shared concept through the corresponding LLM, simple filtering of concepts is executed to check if generated concepts are non-specific, namely "Object", "Thing", "Verb", "Adjective", "Noun", or "Word". If so, high-level concept guidance is omitted (this is only the case for three out of eleven visual classification benchmarks, see also §4). We then augment the default CLIP prompt to "A photo of a {concept}: a {c}." and for WaffleCLIP, the prompt is extended to "A photo of a {concept}: a {c}, which (is/has/etc) {random_sequence}." While the prompt style can likely be improved, this naive extension already offers remarkable benefits.

4. Experiments

We first provide implementation details before comparing WaffleCLIP to DCLIP in §4.2. Extending our observations in Tab. 1, we study the source of performance gains via LLM-generated descriptors (§4.3) and present a better way for introducing semantics into the retrieval process while tackling semantic ambiguities with automatically extracted high-level concepts (§4.4).

In addition to that, §4.5 provides further insights on other benchmarks not studied in e.g. [36], alongside benchmarks measuring out-of-distribution (OOD) generalization. §4.6 showcases a comparison to hand-crafted prompt ensembles, and §4.7 a study into the relationship of WaffleCLIP and latent space noise. §4.8 then provides ablational studies to WaffleCLIP. Finally, further experiments and particularly experimental details and results are included in the supplementary materials.

ViT-B/32	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
CLIP [47]	54.71	62.01	51.28	40.78	39.12	82.59	85.06	43.18	57.34
+ Concepts	↓	↓	52.23	48.86	39.31	84.66	86.73	↓	58.96
DCLIP [36]	55.82	63.12	52.47	43.29	40.47	82.79	86.54	**43.99**	58.56
`WaffleCLIP` (ours)	**55.92** ±0.08	**63.31** ±0.09	**52.38** ±0.12	44.31 ±1.07	40.56 ±0.07	83.25 ±0.21	85.70 ±0.25	43.16 ±0.25	58.57 ±0.41
+ Concepts	↓	↓	52.83 ±0.19	48.51 ±0.70	40.97 ±0.08	**85.21** ±0.06	87.52 ±0.10	↓	59.47 ±0.42
+ GPT descr. + Concepts	↓	↓	**52.77** ±0.26	**51.64** ±0.25	**41.35** ±0.09	84.87 ±0.05	**87.71** ±0.18	↓	**60.21** ±0.20

Table 2: **Image classification with `WaffleCLIP`** which extends input prompts with random word and character sequences and matches the performance of DCLIP [36] using GPT-generated class descriptors. Additional semantic context through high-level concepts (+ *Concepts*) can offer further boosts, particularly on benchmarks where classnames can be generic or ambiguous. We further find that `WaffleCLIP` complements the use of GPT-generated descriptors (+ *GPT descr.*). (↓) denotes same results as previous lines where high-level concept guidance is not applicable. For ViT-L/14 and RN50, see Supp.

4.1. Experimental details

We utilize CLIP [47] as the underlying VLM for `WaffleCLIP`. As there is no direct cost associated with generating random character or word sequences, their number is only bounded by inference speed requirements (which is minimal as all respective language embeddings can be computed *a priori* [36]). However, we find diminishing returns for very high numbers (see also §4.8), and use 30 random descriptors per class (or 15 random character and word descriptor pairs) if not mentioned otherwise, with similar performance for both half or double the descriptor count (c.f. §4.8). All experiments use PyTorch [44] and are conducted on a single NVIDIA 3090Ti GPU. Wherever necessary, fine-grained LLM-generated descriptors are either taken from or generated following the codebase provided by [36]. If not mentioned explicitly, all results involving `WaffleCLIP` are computed over seven random seeds.

Benchmarks. The datasets considered are (mostly from [36]) ImageNet [14] and ImageNetV2 [29], CUB200-2011 [57] (fine-grained bird classification), EuroSAT [23] (satellite image recognition), Places365 [65], Food101 [4], Oxford IIIT Pets [43], DTD (Textures, [12]), Flowers102 [40], FGVCAircraft [34], and Stanford Cars [30].

High-level concepts. Following §3.3, the GPT-3 generated high-level concept for CUB200-2011 is `"Bird"`, `"Land Use"` for EuroSAT, `"Place"` for Places365, `"Food"` for Food101 and `"Breed"` for Oxford Pets. For additional benchmarks, extracted concepts are noted in section §4.5. For ImageNet (V2) and DTD, the concepts are too generic and thus filtered out (`"Object"`, `"Noun"`, or `"Adjective"`), with high-level guidance omitted.

4.2. `WaffleCLIP` vs LLM-generated descriptors

We start by analyzing the impact of randomization beyond fixed, randomized sets of fine-grained LLM-generated descriptors as done in Tab. 1, by instead using randomized character or word descriptors through our proposed `WaffleCLIP`. For that, we investigate visual classification accuracies across the eight diverse benchmarks studied in [36] in Tab. 2, where we compare `WaffleCLIP`, which does not use any external LLMs, with DCLIP.

We find that averaging over randomized descriptors yields performances comparable to or better than those obtained with LLM-generated fine-grained descriptors over a majority of studied datasets, with average performance being comparable: 58.56% using DCLIP versus 58.57% for `WaffleCLIP` with a ViT-B/32 backbone and (see supp. material) $69.14\% \rightarrow 68.95\%$ for ViT-L/14, and $54.77\% \rightarrow 54.20\%$ for ResNet50. Beyond the inherently zero-shot nature of `WaffleCLIP` and ease of use, these results highlight that improved visual classification with pretrained VLMs does not require external LLMs, and further cements prompt averaging as a potential key driver behind DCLIP.

4.3. Are descriptors from LLMs obsolete?

Our results above question the benefits of LLM-generated fine-grained semantics, as averaging over fully randomized character and word sequences achieves comparable performance. But does that mean that there is no benefit in leveraging descriptors produced by LLMs?

4.3.1 Impact of Descriptor Averaging

To better understand this, we extend our motivational experiments from Tab. 1. First, we look at what happens when not performing averaging over all class descriptor distances as in DCLIP, but instead choosing the maximum. If additional fine-grained semantics were indeed beneficial, selecting the most suitable one should similarly raise the performance. However, as Tab. 3 reveals, performance actually drops, showing that the VLM cannot leverage the additional semantics to improve visual classification performance[2]. Instead, this again points to descriptor ensembling *as the main driver in performance*.

We further support this by studying additional descriptor randomization variants beyond those in §3.2. In particu-

[2] This is potentially influenced by bag-of-words behavior of CLIP-like Vision-Language models as studied e.g. in [63]. We leave the detailed analysis of this to future research.

ViT-B/32	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	Flowers102	FGVCAircraft	Stanford Cars	Avg
CLIP [47]	54.71	62.01	51.28	40.78	39.12	82.59	85.06	43.18	62.89	24.99	58.54	55.01
DCLIP [36] (*mean*)	**55.82**	**63.12**	**52.47**	**43.29**	**40.47**	**82.79**	86.54	**43.99**	**64.01**	**26.94**	**57.08**	**56.05**
DCLIP [36] (*max*)	54.41	61.67	52.40	37.11	37.21	82.37	**88.03**	43.35	63.62	25.77	56.21	54.74

Table 3: **Importance of semantics in DCLIP.** The favorable performance of similarity score averaging (*mean*) over simply selecting the maximum similarity score (*max*), which in some cases can even underperform the CLIP baseline built upon, points to the limited impact of LLM-generated semantics on improved visual classification.

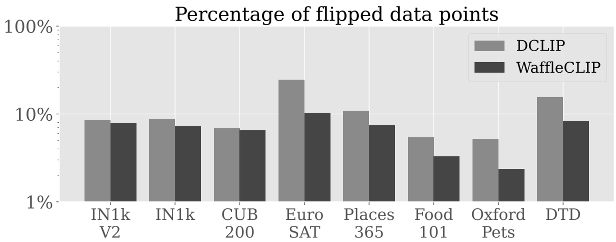

Figure 3: Label flipping experiment from CLIP to DCLIP or WaffleCLIP. Each bar indicates the percentage of data points getting either positively or negatively flipped (i.e. labelled correctly or incorrectly adjusted) when switching from CLIP to either DCLIP or WaffleCLIP. The consistently higher flip percentage indicates structural differences between natural language descriptors and randomized ones.

lar, instead of swapping specific descriptors, we interchange full class-specific descriptor lists (*interchanged*). As descriptions often contain class-specific keywords, this models a systematic semantic shift away from the actual class. Additionally, we evaluate shuffling words within a descriptor list (*shuffled*), and descriptor lists subsampled from all available ones (*random*). This gives a progression from systematic to more independent descriptor randomization.

Our results in Tab. 4 reveal that directly interchanging full *class-dependent* descriptor lists (*interchanged*) drops performance significantly (e.g. from 58.56% to 55.03% for ViT-B/32). In cases where no such shift is happening, we find performances to match DCLIP (e.g. 86.54% → 86.28% on Oxford Pets). Similarly, when moving from a systematic shift closer to fully randomized descriptors, performance approaches DCLIP (*scrambled* with 58.56% → 57.55% to *random* with 58.56% → 58.02%, see supp. material for more results).

While this offers further evidence for WaffleCLIP and the fact that class-dependent ensembling drives gains, it does not yet allow us to directly compare the impact on the prediction behavior of LLM-generated descriptors and randomized ones.

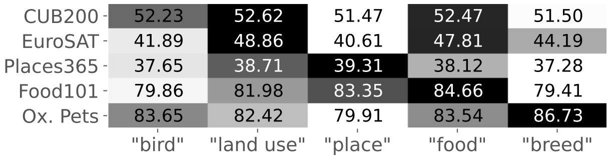

Figure 4: Study of the semantic impact of GPT-3 generated high-level concepts through a semantic confusion matrix, where we cycle high-level concepts between each benchmark. We find that interchanging the concepts generally reduces performance, indicating that high-level concepts provide complementary semantic context.

4.3.2 Structural differences between LLM-generated and randomized descriptors

We consider the percentages of samples that get positively or negatively flipped - i.e. classified correctly while previously being classified incorrectly (and vice versa) - when moving from CLIP to either DCLIP or WaffleCLIP in Fig. 3. We find that using LLM-generated fine-grained descriptors flips significantly more predictions than randomized words and characters, even when WaffleCLIP outperforms DCLIP. For example, DCLIP achieves 43.29% compared to WaffleCLIP with 44.31% on EuroSAT or 82.79% to 83.25% on Food101 in Tab. 2, but DCLIP flips a significantly larger portion of samples than WaffleCLIP.

This reveals that full sentence, LLM-generated descriptors have a *structurally different* impact on the classification process, which we find to be *complementary* to randomization (see Tab. 2, + *GPT descr.*), where the use of both leads to additional improvements over WaffleCLIP (e.g. 58.57% → 60.21% for ViT-B/32).

This means that even if additional semantics are not the leading factor, LLMs for structured descriptor generation can still facilitate more robust class embeddings. Even *with* access to an external model for producing class descriptors, WaffleCLIP can provide additional benefits.

4.4. Semantic guidance with high-level concepts

While we verified the relevance of additional semantic context through fine-grained descriptors, methods using additional fine-grained class information [36, 42, 46] suffer from the inherent ambiguities of some class names. As proposed in §3.3, our aim is to understand if high-level semantic context can be used to resolve such ambiguities by providing coarse semantic guidance for the class-retrieval process.

Our results with extracted high-level concepts in Tab. 2, i.e. (+ *Concepts*), demonstrate consistent and significant improvements across most benchmarks and backbones

ViT-B/32	ImageNetV2	ImageNet	CUB200	EuroSAT	Places365	Food101	Oxford Pets	DTD	**Avg**
DCLIP [36]	**55.82**	**63.12**	52.47	**43.29**	40.47	82.79	86.54	**43.99**	**58.56**
DCLIP (interchanged)	52.51 ±0.42	59.62 ±0.13	52.52 ±0.41	33.63 ±4.16	35.52 ±0.32	81.71 ±0.35	86.28 ±0.50	38.42 ±1.14	55.03 ±1.56
DCLIP (scrambled)	55.12 ±0.12	62.57 ±0.12	52.18 ±0.28	40.48 ±2.52	39.91 ±0.08	82.46 ±0.13	86.10 ±0.40	41.58 ±0.31	57.55 ±0.92
DCLIP (random, 1x)	54.11 ±0.28	61.37 ±0.18	52.42 ±0.19	36.83 ±4.27	38.80 ±0.26	82.86 ±0.23	85.99 ±0.62	42.20 ±0.85	56.82 ±1.57
DCLIP (random, 5x)	55.43 ±0.12	62.81 ±0.05	52.66 ±0.17	38.57 ±1.52	40.54 ±0.05	84.03 ±0.11	86.75 ±0.21	43.41 ±0.74	58.02 ±0.61

Table 4: **Progression from systematic to fully randomized descriptor scrambling.** To model systematic semantic shifts, we randomly swap descriptor lists between classes (*interchanged*), before progressing to shuffling descriptor words within the classes (*scrambled*) and randomly sampling LLM-generated descriptors for each class (*random*) from the complete set of descriptors with counts as in (or five times that of) DCLIP (*1x*, *5x*). As can be seen, a systematic shift results in a notable performance drop, while more independently randomized descriptors can recover the DCLIP performance, aligning with the observation that fully randomized prompt averaging is the main performance driver for WaffleCLIP.

ViT-B/32	Flowers102	FGVCAircraft	Stanford Cars	**Avg**
CLIP [47]	62.89	24.99	58.54	48.81
DCLIP [36]	64.01	26.94	57.08	49.34
WaffleCLIP	66.27 ±0.26	25.66 ±0.19	58.91 ±0.17	50.28 ±0.21
+ Concepts	**67.19** ±0.19	28.44 ±0.22	**59.70** ±0.12	**51.78** ±0.18
+ GPT dsc. + Conc.	66.71 ±0.39	**28.96** ±0.37	59.33 ±0.14	51.67 ±0.32

Table 5: We find similar performance improvements with WaffleCLIP and high-level concept guidance for three additional standard visual classification benchmarks not studied in [36], which in parts do not benefit from LLM-generated descriptors (e.g. *Stanford Cars*).

Benchmarks	ImageNet-R [24]	ImageNet-S [58]	ImageNet-A [25]
CLIP [47]	65.97	40.73	29.63
DCLIP [36]	65.12	41.09	29.19
WaffleCLIP	**67.31**	**42.00**	**31.52**

Table 6: Performance gains of WaffleCLIP on distribution-shifted datasets to study impacts on OOD generalization. Our results further highlight the general applicability of simple averaging over randomized descriptors, even in cases where the use of natural language ones may fail.

when used with CLIP, with WaffleCLIP, and even alongside WaffleCLIP and DCLIP. These improvements are especially evident on benchmarks with ambiguous (e.g. Food101) or generic labeling (e.g. EuroSAT, with labels such as *Industrial* or *Residential*): For ViT-B/32, classification accuracy increases from 40.78% to 48.86% when applied to CLIP.

Overall, the average classification accuracy also increases consistently (e.g. from 57.34% to 58.96% for ViT-B/32). This even beats DCLIP, despite only being applicable on five out of eight benchmarks (58.96% versus 58.56%). When applied to WaffleCLIP, improvements across most benchmark and backbone settings are also significant, although we find diminishing returns on the largest backbone, ViT-L/14, with average performance increasing only from 68.95% to 69.12% (see suppl. material). This might be due to its capabilities of retaining the most common concepts associated with specific classes, resulting in a robust class retrieval setup when averaging over multiple randomized descriptor variants.

We verify the benefits of high-level semantics further by looking at performance changes when concepts are interchanged (Fig. 4). For most benchmarks, the largest improvements are obtained with GPT-generated concepts. Some off-diagonal terms with higher scores, e.g. CUB200 where "bird" performs similar to/worse than "land use"/"food", do appear out of distribution and warrant future research to improve our understanding of how semantics concepts are truly encoded in large VLMs.

However, seeing maximum performances primarily on the diagonal heuristically supports that additional semantics introduced as high-level concepts and commonalities, can offer reliable guidance. Indeed, considering a selection of ambiguous samples such as "Boxer" or "Sphynx" in the Oxford Pets dataset, "Mussels", "Oysters" or "Grilled Salmon" in the Food101 dataset, or highly generic labels such as "Industrial" or "Residential" in the EuroSAT satellite image dataset, we find a consistent increase in average similarity to all associated test images by up to 13%. This confirms that concept guidance can re-align and refine class embeddings based on the relevant context.

4.5. Evaluation on additional (OOD) benchmarks.

We observe further evidence for the generality of WaffleCLIP and concept guidance by studying three additional benchmarks beyond those in Tab. 2 and [36]: Flowers102 [40] (extracted concept: "flower"), FGVCAircraft [34] ("aircraft"), and StanfordCars [30] ("car"). Our results in Tab. 5 (and in the suppl. material for other backbones) again show consistent gains when going from CLIP to WaffleCLIP or WaffleCLIP + *Concepts*.

Interestingly, DCLIP is detrimental on very fine-grained benchmarks like Stanford Cars, losing 1.46% against CLIP. We speculate that this is due to semantically similar descriptors for multiple classes that are coarser than the actual class label (e.g. "BMW Active Hybrid" and "BMW 1 Series" being assigned similar generic BMW descriptors).

ViT-B/32	IN1k-V2	IN1k	CUB	Euro	Places	Food	Pets	DTD	Flowers	FGVC	Cars	**Avg**
CLIP [47]	54.71	62.01	51.28	40.78	39.12	82.59	85.06	43.18	62.89	24.99	58.54	55.01
DCLIP [36]	55.82	63.12	52.47	43.29	40.47	82.79	86.54	**43.99**	64.01	26.94	57.08	56.05
P. Ensemble	55.49 ±0.21	62.79 ±0.29	51.46 ±0.43	45.76 ±0.49	40.58 ±0.06	82.67 ±0.37	83.26 ±0.72	42.53 ±0.54	63.30 ±0.33	25.14 ±0.45	58.38 ±0.29	55.58 ±0.42
+ Concepts	↓	↓	**52.08** ±0.17	**49.80** ±0.66	**40.61** ±0.14	**84.45** ±0.15	**87.42** ±0.20	↓	**65.38** ±0.27	**26.64** ±0.50	**59.12** ±0.14	**56.94** ±0.34
`WaffleCLIP`	**55.92** ±0.08	**63.31** ±0.09	52.38 ±0.12	44.31 ±1.07	40.56 ±0.07	83.25 ±0.21	85.70 ±0.25	43.16 ±0.25	66.27 ±0.26	25.66 ±0.19	58.91 ±0.17	56.31 ±0.37
+ Concepts	↓	↓	**52.83** ±0.19	**48.51** ±0.70	**40.97** ±0.08	**85.21** ±0.06	**87.52** ±0.10	↓	**67.19** ±0.19	**28.44** ±0.22	**59.70** ±0.12	57.52 ±0.26

Table 7: **Prompt ensembling versus `WaffleCLIP` (+concepts)**. Across all visual classification benchmarks, we conduct comparisons to equivalent prompt ensembling, which leverages handcrafted prompts. Our results show matching or improved performance of `WaffleCLIP` (improving on eight out of eleven benchmarks), with the increase in average classification performance of `WaffleCLIP` compared to prompt ensembling higher than the increase of a prompt-ensembled version over the respective standard CLIP baseline, **without** requiring a handcrafted list of prompts.

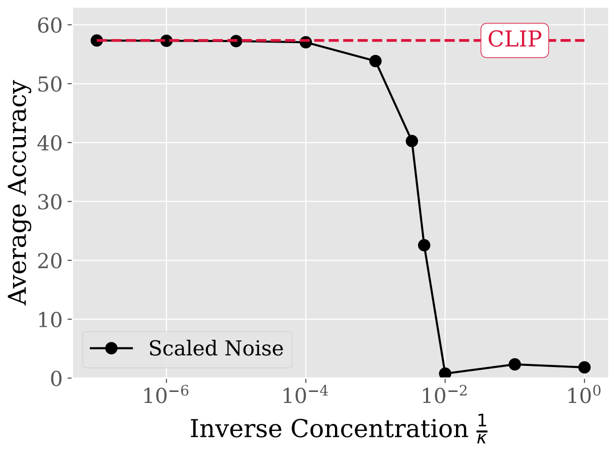

Figure 5: We evaluate changes in the performance of the CLIP baseline when using latent space noise (modeled using vMF distributions centered around the default, unaltered CLIP language embeddings) and thus also provide an implicit comparison against `WaffleCLIP`). However, as can be seen, reducing latent noise (i.e. increasing concentration κ) converges to initial performance while significantly dropping the performance when increasing the latent noise scale. This highlights no notable benefit of deploying noise in the latent space.

Consequently, embeddings of related classes are systematically moved too close, harming performance. Meanwhile, `WaffleCLIP` (+ *Concepts*) can still offer performance boosts (58.54% → 58.91% → 59.70%).

Furthermore, we study `WaffleCLIP` on benchmarks measuring OOD generalization: Adversarial natural images (ImageNet-A, [25]), sketches (ImageNet-S, [58]) and renditions (ImageNet-R, [24]). Results in Tab. 6 show that while DCLIP does not improve consistently, `WaffleCLIP` operates well even for out-of-distribution data (e.g. 29.63% → 31.52% on ImageNet-A).

4.6. Comparison to prompt ensembles

We also compare `WaffleCLIP` to prompt ensembling (c.f. e.g. [46]) with the same budget of 30 randomly selected prompt options from a list of eighty handcrafted ones (taken from [46], such as `"A tattoo of a {class}."`, `"A {class} in a video game."`, ...). Unlike `WaffleCLIP`, prompt ensembling still requires human input and design. Results on all eleven benchmarks are listed in Tab. 7, which favor `WaffleCLIP`, outperforming prompt ensembling in eight out of eleven benchmarks and comparable performance on the remaining ones.

In particular, improvements over prompt ensembling are higher than the improvement of prompt ensembling over vanilla CLIP (56.31% → 55.58% → 55.01%). This further supports the benefit of extracting more robust semantic representations, for which randomized descriptors provide a cheap and suitable tool.

In addition to that, we highlight the complementarity of high-level concept guidance in combination with prompt ensembling in Tab. 7 (wherever the classname is included, we simply use `"a {concept}: a {classname}"` instead), raising the average classification accuracy from 55.58% to 56.94%.

4.7. Comparison to latent noise

To highlight that input-level class-conditioned randomization is crucial, we compare our results using `WaffleCLIP` to latent random noise applied on the hyperspherical representation of CLIP. To model the corresponding noise distribution, we use unimodal von-Mises-Fisher distributions (as commonly done, c.f. e.g. [59, 13, 68, 50]) of class embedding vectors $\hat{\phi}^c$:

$$p(\hat{\phi}^c|\phi_l^c, \kappa) = \mathcal{C}_d(\kappa)\exp(\kappa{\phi_l^c}^T\hat{\phi}^c), \tag{3}$$

centered around a class centroid vectors ϕ_l^c with constant normalization $\mathcal{C}_d(\kappa)$ only dependent on the dimensionality of ϕ and concentration κ. Note that ϕ_l^c is simply the language embedding produced by CLIP when utilizing the unaltered classname and input prompt. To sample from a vMF distribution around each class embedding, we leverage the sampler utilized in [13, 27] with the same budget of 30 noise embeddings.

Average performance as a function of the (inverse) concentration κ is visualized in Fig. 5. For high concentrations (i.e. random embedding samples placed close to the mean direction), one can replicate the CLIP performance. For higher variances, performance continuously drops, with a hard inflection at around $\kappa \approx 10^4$. This shows that class-conditioned randomized descriptors as used in WaffleCLIP are crucial for providing a more robust estimate of semantic concepts, and cannot be simulated through simple embedding space noise.

4.8. Ablations

4.8.1 Dependence on descriptor counts

We study the impact of the randomized word and character sequence pair count for WaffleCLIP in Fig. 6. A value of one indicates a single pair comprising a random words and a separate random characters descriptor, respectively. As can be seen, we already achieve competitive performance with 4 to 15 descriptor pairs (c.f. DCLIP in Tab. 2), while consistently outperforming CLIP (blue line) even with a single randomized descriptor pair. As class embeddings can be computed *a priori*, and the generation of random words and characters does not require external model queries, the impact on overall setup and inference time is low, making WaffleCLIP and its extensions very attractive for enhancing image classification performance of VLMs.

4.8.2 Impact of randomization types

Finally, we analyze how performance changes when either using only random character sequences or only random word sequences instead of a combination of both as in WaffleCLIP. Across benchmarks and architectures (see Tab. 8), we observe dichotomies in performance between either random word or random character sequences, often performing either best or worst on a specific benchmark and backbone, while the joint usage of random words and character sequences strikes a consistent and best transferable average improvement across benchmarks and backbone architectures. Therefore, we chose the joint usage of both random words and characters as our default setup.

Avg.	ViT-B/32	ViT-L/14	RN50
Joint	58.57 ±0.41	68.95 ±0.18	54.20 ±0.23
Random Words	58.18 ±0.44	68.73 ±0.58	55.24 ±0.41
Random Characters	58.59 ±0.27	68.02 ±0.14	53.79 ±0.16

Table 8: **Randomized descriptor modes.** We provide a quick comparison between the joint usage of randomized word and character sequences as opposed to their singular usage, i.e. either only randomized word or character sequences. Our experiments show that joint usage provides the most consistent performance improvements across benchmarks and backbones.

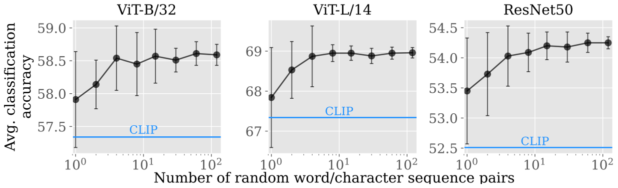

Figure 6: Ablation study on the number of randomized word and character descriptors used in WaffleCLIP. We find consistent competitive performance gains with just four randomized descriptor pairs (both random words and random characters, c.f. DCLIP Tab. 2). Note that the CLIP reference (blue line) is already outperformed with just a single descriptor pair.

5. Conclusion

In this work, we systematically examined the benefits of using LLM-generated class descriptors for improved training-free image classification with vision-language models (VLMs). In-depth studies reveal how similar performance gains can be achieved by replacing LLM-generated descriptors with randomized ones, giving rise to WaffleCLIP. Without access to external LLMs, across eleven visual classification benchmarks, we get comparable or better results than those obtained when using fine-grained GPT-3 generated descriptors. This makes WaffleCLIP very attractive for practical use in true zero-shot scenarios, and it serves as a crucial sanity check for future methods using external queries. We also show that VLMs struggle to leverage the actual semantics introduced through fine-grained semantic descriptors, and instead show that if given access to external LLMs, semantics are better exploited through coarse, high-level concepts. Using specific queries, we show how these can be automatically extracted, while jointly helping to address issues of class ambiguity.

Acknowledgements

This work was supported by DFG project number 276693517, by BMBF FKZ: 01IS18039A, by the ERC (853489 - DEXIM), and by EXC number 2064/1 – project number 390727645. KR and JMK thank the European Laboratory for Learning and Intelligent Systems (ELLIS) PhD program and the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for support.

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv:2203.17274*, 2022.

[2] Hritik Bansal and Aditya Grover. Leaving reality to imagination: Robust classification via generated datasets. *arXiv:2302.02503*, 2023.

[3] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. In *ICLR*, 2018.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *ECCV*, 2014.

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020.

[6] Sebastian Bujwid and Josephine Sullivan. Large-scale zero-shot image classification from rich and diverse textual descriptions. In *Workshop on Beyond Vision and LANguage: inTEgrating Real-world kNowledge*, 2021.

[7] Aochuan Chen, Yuguang Yao, Pin-Yu Chen, Yihua Zhang, and Sijia Liu. Understanding and improving visual prompting: A label-mapping perspective. *arXiv:2211.11635*, 2022.

[8] Guangyi Chen, Weiran Yao, Xiangchen Song, Xinyue Li, Yongming Rao, and Kun Zhang. PLOT: Prompt learning with optimal transport for vision-language models. In *ICLR*, 2023.

[9] Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *ACL*, 2020.

[10] Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. Towards robust neural machine translation. In *ACL*, 2018.

[11] Subhabrata Choudhury, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. The curious layperson: Fine-grained image recognition without expert labels. In *BMVC*, 2021.

[12] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014.

[13] Tim R. Davidson, Luca Falorsi, Nicola De Cao, Thomas Kipf, and Jakub M. Tomczak. Hyperspherical variational auto-encoders. *Conference on Uncertainty in Artificial Intelligence*, 2018.

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[15] Lisa Dunlap, Clara Mohri, Devin Guillory, Han Zhang, Trevor Darrell, Joseph E. Gonzalez, Aditi Raghunathan, and Anja Rohrbach. Using language to extend to unseen domains, 2023.

[16] Mohamed Elhoseiny, Yizhe Zhu, Han Zhang, and Ahmed Elgammal. Link the head to the" beak": Zero shot learning from noisy text description at part precision. In *CVPR*, 2017.

[17] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. In *ACL-IJCNLP*, 2021.

[18] Yanming Guo, Yu Liu, Erwin M Bakker, Yuanhao Guo, and Michael S Lew. Cnn-rnn: a large-scale hierarchical image classification framework. *Multimedia tools and applications*, 2018.

[19] Xiaoshuai Hao, Yi Zhu, Srikar Appalaraju, Aston Zhang, Wanqian Zhang, Bo Li, and Mu Li. Mixgen: A new multi-modal data augmentation. In *WACV*, 2023.

[20] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and XIAOJUAN QI. Is synthetic data from generative models ready for image recognition? In *ICLR*, 2023.

[21] Xiangteng He and Yuxin Peng. Fine-grained image classification via combining vision and language. In *CVPR*, 2017.

[22] Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. How robust are character-based word embeddings in tagging and mt against wrod scramlbing or randdm nouse? In *Association for Machine Translation in the Americas*, 2018.

[23] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2017.

[24] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.

[25] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021.

[26] Tony Huang, Jack Chu, and Fangyun Wei. Unsupervised prompt learning for vision-language models. *arXiv:2204.03649*, 2022.

[27] Michael Kirchhof, Karsten Roth, Zeynep Akata, and Enkelejda Kasneci. A non-isotropic probabilistic take on proxy-based deep metric learning. In *ECCV*, 2022.

[28] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *NAACL-HLT*, 2018.

[29] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better? In *CVPR*, 2019.

[30] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *IEEE Workshop on 3D Representation and Recognition*, 2013.

[31] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 2023.

[32] Jochem Loedeman, Maarten C Stol, Tengda Han, and Yuki M Asano. Prompt generation networks for efficient adaptation of frozen vision transformers. *arXiv:2210.06466*, 2022.

[33] Yuning Lu, Jianzhuang Liu, Yonggang Zhang, Yajing Liu, and Xinmei Tian. Prompt distribution learning. In *CVPR*, 2022.

[34] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv:1306.5151*, 2013.

[35] Chengzhi Mao, Revant Teotia, Amrutha Sundar, Sachit Menon, Junfeng Yang, Xin Wang, and Carl Vondrick. Doubly right object recognition: A why prompt for visual rationales, 2023.

[36] Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *ICLR*, 2023.

[37] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998.

[38] Muhammad Ferjad Naeem, Muhammad Gul Zain Ali Khan, Yongqin Xian, Muhammad Zeshan Afzal, Didier Stricker, Luc Van Gool, and Federico Tombari. I2mvformer: Large language model generated multi-view document supervision for zero-shot image classification. In *CVPR*, 2023.

[39] Muhammad Ferjad Naeem, Yongqin Xian, Luc Van Gool, and Federico Tombari. I2dformer: Learning image to document attention for zero-shot image classification. In *NeurIPS*, 2022.

[40] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[41] Xing Niu, Prashant Mathur, Georgiana Dinu, and Yaser Al-Onaizan. Evaluating robustness to input perturbations for neural machine translation. In *ACL*, 2020.

[42] Zachary Novack, Saurabh Garg, Julian McAuley, and Zachary C Lipton. Chils: Zero-shot image classification with hierarchical label sets. *arXiv:2302.02551*, 2023.

[43] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012.

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019.

[45] Tzuf Paz-Argaman, Reut Tsarfaty, Gal Chechik, and Yuval Atzmon. Zest: Zero-shot learning from text descriptions using textual similarity and visual summarization. In *EMNLP*, 2020.

[46] Sarah Pratt, Rosanne Liu, and Ali Farhadi. What does a platypus look like? generating customized prompts for zero-shot image classification. *arXiv:2209.03320*, 2022.

[47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[48] Scott Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. Learning deep representations of fine-grained visual descriptions. In *CVPR*, 2016.

[49] Karsten Roth, Oriol Vinyals, and Zeynep Akata. Integrating language guidance into vision-based deep metric learning. In *CVPR*, 2022.

[50] Karsten Roth, Oriol Vinyals, and Zeynep Akata. Non-isotropy regularization for proxy-based deep metric learning. In *CVPR*, 2022.

[51] Gözde Gül Şahin. To augment or not to augment? a comparative study on text augmentation techniques for low-resource nlp. *Computational Linguistics*, 2022.

[52] Sheng Shen, Chunyuan Li, Xiaowei Hu, Yujia Xie, Jianwei Yang, Pengchuan Zhang, Anna Rohrbach, Zhe Gan, Lijuan Wang, Lu Yuan, et al. K-lite: Learning transferable visual models with external knowledge. *arXiv:2204.09222*, 2022.

[53] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 2019.

[54] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. *arXiv:2209.07511*, 2022.

[55] Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, S Yu Philip, and Lifang He. Mixup-transformer: Dynamic data augmentation for nlp tasks. In *Computational Linguistics*, 2020.

[56] Vishaal Udandarao, Ankush Gupta, and Samuel Albanie. Sus-x: Training-free name-only transfer of vision-language models. *arXiv:2211.16198*, 2022.

[57] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[58] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. In *Advances in Neural Information Processing Systems*, pages 10506–10518, 2019.

[59] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020.

[60] Jason Wei and Kai Zou. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *EMNLP-IJCNLP*, 2019.

[61] Junyang Wu, Xianhang Li, Chen Wei, Huiyu Wang, Alan Yuille, Yuyin Zhou, and Cihang Xie. Unleashing the power of visual prompting at the pixel level. *arXiv:2212.10556*, 2022.

[62] Yinghui Xing, Qirui Wu, De Cheng, Shizhou Zhang, Guoqiang Liang, and Yanning Zhang. Dual modality prompt tuning for vision-language pre-trained model. *arXiv:2208.08340*, 2022.

[63] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *The Eleventh International Conference on Learning Representations*, 2023.

[64] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018.

[65] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE TPAMI*, 2017.

[66] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *CVPR*, 2022.

[67] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *IJCV*, 2022.

[68] Roland S. Zimmermann, Yash Sharma, Steffen Schneider, Matthias Bethge, and Wieland Brendel. Contrastive learning inverts the data generating process. In *ICML*, 2021.