Class-Mixed Diffusion Augmentation for Shortcut-Breaking in Continual Learning

Abhinab Acharya[1] **Dayou Yu**[2] **Qi Yu**[1,3] **Xumin Liu**[1*]
[1]Rochester Institute of Technology [2]The University of Memphis [3]Amazon Inc.
aa7514@rit.edu dyu@memphis.edu qi.yu@rit.edu xmlics@rit.edu

Abstract

Zero-shot pre-trained diffusion models can provide a training-free source of synthetic image generation for Exemplar-Free Class Incremental Learning (EFCIL). However, same-class generation is limited by the distribution mismatch with real data. Parameter-allocation-based EFCIL can minimize the forgetting in model parameters, yet the performance degrades because task-id prediction fails when task-specific feature extractors overcompress and converge to shortcut-dominant representations. We propose *Class-Mixed Diffusion Augmentation* (CMDA) that uses diffusion models not only as past-task data generators but also as controlled generators that challenge the shortcut reliance. We sample candidate tokens from the vocabulary and select *shortcut-breaking* tokens via classifier-guided filtering, then treat the selected tokens as auxiliary classes during training. We provide theoretical analysis linking shortcut sharing to an intrinsic lower bound on task-id error and show that, under stated assumptions, minimum-CE selection recovers shortcut-breaking tokens with nonzero probability. Experiments on Split-CIFAR100, Split-ImageNet, and Split-CUB show consistent gains over EFCIL baselines, with modest overhead over vanilla diffusion augmentation and far lower cost than diffusion fine-tuning or gradient-guided sampling. Our ablation study further shows that auxiliary-class augmentation contributes most to the performance gain, while classifier-guided selection provides a smaller, consistent improvement.

1 Introduction

Pretrained text-conditioned diffusion models have recently emerged as powerful *training-free* generators that can synthesize diverse images from simple prompts [26, 29, 3, 34]. This capability has made diffusion models attractive for data augmentation, including in continual learning (CL), where synthetic data can partially compensate for limited access to past samples [8, 21, 35, 14, 15]. However, prior work also shows that naive diffusion augmentation can be misaligned with the target data distribution and may even hurt performance, especially for fine-grained recognition [21, 35]. To address this, existing diffusion-based CL methods often resort to training a diffusion model from scratch [8] or fine-tuning it via LoRA and domain adaptation [21], or textual inversion [18]. However, this adds substantial computational cost and can induce overfitting [35]. Thus, the use of a zero-shot pre-trained diffusion model to generate samples for CL has recently increased in popularity and importance [15, 14, 35]. Some methods generate additional *same-class* or *auxiliary-class* samples and train the CL model on a mixture of real and synthetic data [15, 14]. An alternative is to incorporate classifier feedback during sampling, but gradient-guided diffusion requires backpropagation through the denoising process and thus carries high overhead that does not scale well beyond few-shot settings [8, 35].

*Work not related to the position at Amazon.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

We focus on *exemplar-free class-incremental learning* (EFCIL), where storing past data is not allowed [39, 23, 27]. An effective family of EFCIL approaches is *parameter allocation*, which assigns task-specific parameters while sharing some subset of parameters across tasks to preserve learned knowledge [30, 19, 28, 27]. Because past task parameters are frozen, forgetting is largely mitigated at the parameter level. Yet overall performance can still collapse at inference time, since the task identity is unknown and must be inferred to select the appropriate task-specific parameters. Common task-id prediction heuristics (*e.g.,* maximum-logit) are highly sensitive to inter-task feature overlap [19, 28].

We argue that a central cause of task-ID prediction failures is *shortcut-collapse*, in which a task model becomes overly reliant on cues such as background, texture, or style when trained on the current-task objective. These cues, which we call shortcuts, help solve the current task but are insufficient to identify which task an image belongs to. If multiple tasks share similar shortcut distributions, their representations become indistinguishable, making task-id prediction fundamentally difficult. We formalize this effect and show that under *shortcut-collapse*, the Bayes-optimal task-id error is lower bounded by the size of the largest shortcut-sharing group (Lemma 4.2). This observation suggests that effective diffusion augmentation for parameter-allocation EFCIL must not only increase sample diversity, but also *break shortcut sharing* to keep feature extractors separable for task-id prediction.

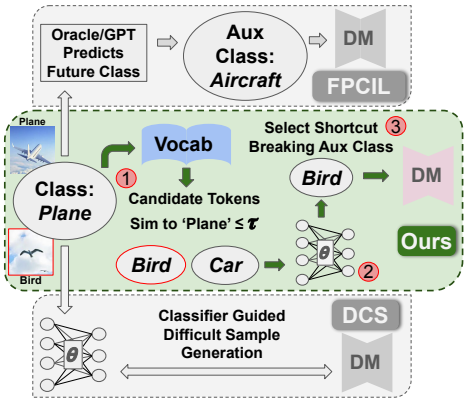


Figure 1: Overview of our method, highlighting the key differences with respect to recent methods. τ controls the similarity of sampled candidate tokens to the class y. "Aux class" refers to auxiliary class, "Vocab" refers to CLIP-Vocabulary, DM refers to Diffusion Model, and $\boldsymbol{\theta}$ refers to CL classifier.

To this end, we propose *Class-Mixed Diffusion Augmentation* (CMDA), a training-free strategy that uses classifier signals *without* backpropagating through the diffusion model. CMDA performs controlled selection of candidate *auxiliary* tokens from CLIP-space similarity regions and scores them using a classifier trained with vanilla diffusion augmentation. Concretely, as visualized in the green/middle section of Figure 1, for each class y in the current task, we **(1)** sample candidate tokens constrained by a similarity threshold τ to avoid near-synonyms, then generate a small number of images per candidate, **(2)** evaluate sampled candidates using the current classifier's Cross-Entropy (CE) loss for label y, and **(3)** select the candidate token with minimum CE loss for class y and assign its generations to a new auxiliary label. CMDA selects tokens that are distinct from class y in text-embedding space but whose generated images receive low CE loss for y under the current classifier. Assigning these images a separate auxiliary labels creates conflicts with the cues used by the classifier, encouraging the model to learn additional task-relevant features. We call a token *shortcut-breaking* if using its generated images as an auxiliary class reduces the dependence between shortcut cues and class labels (Definition 4.4). Low CE serves as a practical selection proxy rather than a guarantee of this property.

As shown in Figure 1, unlike classifier gradient-guided generation method DCS [35], our method CMDA uses classifier feedback only for *token selection* (forward evaluation), retaining the efficiency of vanilla generation at scale. Also, we focus on sampling shortcut-breaking auxiliary classes instead of just sampling from the same class. On the contrary, although FPCIL [14] samples auxiliary classes using Oracle/GPT-based prediction of future classes, such predictions are not aligned with the learned knowledge of the classifier and may not contain the shortcut feature the classifier is relying on. Our method instead performs controlled token-sampling from different similarity regions of the vocabulary and uses the CL classifier to guide the selection of shortcut-breaking tokens. Empirically, we find a consistent region where moderately dissimilar tokens (rather than near-synonyms) produce the most effective shortcut-breaking augmentation (Fig. 3). We find that while auxiliary-class augmentation produces good performance gain, the classifier-guided selection provides additional improvement.

Our contributions are:

1. A training-free diffusion augmentation method (CMDA) that uses classifier loss to select auxiliary tokens and assigns their generations separate labels, without back-propagating through the diffusion model.
2. A theoretical analysis linking shortcut sharing to a lower bound on task-ID error and establishing a probabilistic guarantee for minimum-CE to select shortcut-breaking tokens under stated assumptions.
3. Empirical results showing consistent improvements over competitive EFCIL baselines on Split-CIFAR100, Split-ImageNet, and Split-CUB, with detailed ablations demonstrating the importance of classifier-guided selection and similarity control.

2 Related Works

2.1 Non-Diffusion-Based Methods

Parameter-allocation methods: These approaches mitigate forgetting by assigning task-specific parameters while sharing a subset of parameters across tasks. HAT uses hard attention to selectively utilize parameters for task-incremental learning [30], while TPL [19] trains a separate gating network using a large buffer for task-id prediction. MoE-style methods [1, 37] rely on multiple autoencoders to estimate input-task relevance. Related confidence-based method, such as HAT_{CIL} [17] and MORE [16], infer task-id by selecting the most confident classifier. **Exemplar-free continual learning (EFCIL):** EFCIL methods avoid storing past samples. Prototype-based methods (PASS [39]) retain feature prototypes, while FeTril [23] synthesizes pseudo-features by transforming new class features. Some generate inputs via model inversion to match activations of past data (SSRE [36], ABD [32], R-DFCIL [7]). IL2A [38] reduces feature and classifier bias through augmentation in both feature and label spaces. **Parameter-allocation EFCIL:** AdaGauss [28] adapts prior-task feature distributions via distillation with a frozen previous extractor, while SEED [27] allocates experts per task and allows expert sharing. In contrast, our work leverages *pretrained* diffusion models to synthesize samples. Besides continual learning, methods such as FracMix [13] and Cap2aug [25] can also be used for data augmentation. However, these methods neither identify the shortcuts used by the CL classifier nor assign conflicting labels to those shortcuts. Our method (CMDA) instead imposes an upper bound on token–class similarity, then uses classifier-guided low-CE selection and assigns the generations separate auxiliary labels, creating shortcut–label conflicts and improving task-ID separability.

2.2 Diffusion-Based Methods

Training diffusion models from scratch: DDGR [8] trains a diffusion model on the current task and uses classifier gradients during sampling to reduce cross-entropy for the target class. JDCL [31] repurposes intermediate UNet features to construct a continual learning classifier. **Fine-tuning diffusion models:** DiffClass [21] fine-tunes a diffusion model (e.g., via LoRA) and applies domain adaptation to reduce real–synthetic mismatch. DDDR [18] addresses federated CL using textual inversion to learn class embeddings for later generation. Training the diffusion model from scratch is computationally costly and sometime induce overfitting [35]. For this reason, recent studies also focus on training free or zero-shot generation. **Zero-shot diffusion for generation beyond CL:** Several works study controlled or diversity-improving generation without targeting continual learning, including CADS [29], VSG [3], Boost-and-Skip [34], and Dream-OOD [6]. One of the relevant work, DiffCoRe-Mix [12], is a general, non-continual-learning augmentation method that uses contextual and negative prompts to generate same-class counterparts, retains generations satisfying a lower-bound CLIP-similarity criterion, and mixes them with real images at the pixel or patch level. In contrast, CMDA uses classifier-guided token selection to construct separately labeled auxiliary classes for task-ID prediction. **Zero-shot diffusion for CL:** DCS [35] backpropagates classifier gradient into the sampling process to generate difficult examples for few-shot CIL, but this incurs substantial overhead and is challenging to scale. MADM [15] uses class descriptors in the prompt to generate diverse prior-task samples and combines them with the replay buffer, hence not purely exemplar-free. FPCIL [14] generates images for potential future classes as auxiliary classes, but does not use classifier feedback. Our method also uses auxiliary classes, but crucially *selects* auxiliary tokens using classifier signals from a parameter-allocation-based model to mitigate shortcut-dominant representations, without backpropagating through the diffusion model during generation.

3 Preliminary

Text conditioned Diffusion Model: A pretrained text-conditioned diffusion model can be used to synthesize images conditioned on a text prompt. Stable Diffusion is a latent diffusion model which operates in the latent space of a pretrained variational autoencoder (VAE). Let $\mathrm{Enc}(\cdot)$ and $\mathrm{Dec}(\cdot)$ denote the VAE encoder and decoder, respectively. An image x is mapped to a latent $z_0 = \mathrm{Enc}(x)$, and an image can be reconstructed from a latent by $x = \mathrm{Dec}(z_0)$. Generation proceeds by sampling Gaussian noise $z_{\tilde{n}} \sim \mathcal{N}(0, \mathbf{I})$ and iteratively applying a reverse diffusion process for timesteps $i = \tilde{n}, \tilde{n}-1, \ldots, 0$. Given a text prompt x_p (*e.g.,* a class name possibly augmented with additional tokens), at each timestep i, a denoising network (a U-Net) parameterized by β predicts the noise component: $\hat{\epsilon}_\beta(z_i, i, x_p)$ which is used to compute the next step latent z_{i-1} via a standard sampler (*e.g.,* DDPM):

$$z_{i-1} = \mathrm{Step}\big(z_i,\ i,\ \hat{\epsilon}_\beta(z_i, i, x_p)\big) \tag{1}$$

After $\tilde{n}$ reverse steps, the final latent z_0 is decoded to obtain the image $\tilde{x} = \mathrm{Dec}(z_0)$. This work utilizes prompts that take the form $x_p = \mathrm{concat}(\mathrm{desc}, \mathrm{name})$ where desc is the randomly sampled description (*e.g.,* "a photo of a ") and name is the class name token [15, 14].

Parameter Allocation Based CL: We consider an exemplar-free class incremental learning setting with K tasks. Let $T \in \{1, \ldots, K\}$ denote a random variable for the task identity (task-id) with uniform prior $P(T = t) = 1/K$. For each task t, data are drawn from a task-specific distribution $(X, Y) \mid (T = t) \sim P_t(X, Y)$, where X and Y represent the random variable for input image and corresponding label respectively. Let, $\mathcal{X}_t$ and $\mathcal{Y}_t$ represent the set of images and labels belonging to task t respectively such that $X \in \mathcal{X}_t$ and $Y \in \mathcal{Y}$.

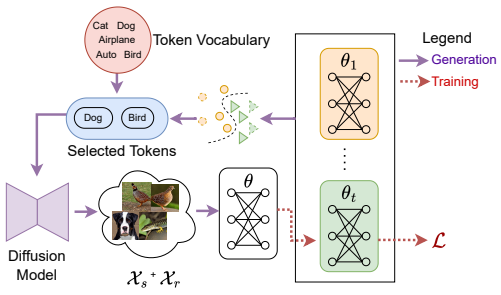


Figure 2: Overall framework. Parameter allocation-based methods use shared parameters θ and task-specific parameters θ_t, and are trained using both synthetic samples $\mathcal{X}_s$ and real samples $\mathcal{X}_r$. Unique to our method, information from the classifiers is used to select the tokens for generation.

The parameter allocation based continual learning method employs combination of shared parameters θ and task-specific parameters θ_t. It also consists of a feature extractor $F_t(X) := F(\theta, \theta_t, X)$ and a classifier $H_t(X) := H(F_t(X), \phi_t)$ parameterized by ϕ_t. During the training of task t, only the data belonging to task t is available and the corresponding task model $H_t(\cdot)$ is trained on $(X, Y) \sim P_t$ by minimizing the standard cross-entropy loss ($\mathcal{L}_{\mathrm{CE}}$): $\min_{\theta, \theta_t, \phi_t} \mathbb{E}_{(X,Y)\sim P_t}\big[\mathcal{L}_{\mathrm{CE}}(H_t(X), Y)\big]$

During inference, the task identity T is unknown and must be identified to select the appropriate task-specific parameter to be used for classification. A common task-id prediction heuristic is *maximum-logit* based selection where given an test input $\tilde{x}$, the task-id of the task-specific parameters that produce the maximum logit for that input is chosen: $\tilde{t} = g(\tilde{x}) := \arg\max_{t \in \{1,\ldots,K\}} \max_{y \in \mathcal{Y}_t} H_t(\tilde{x})_y$ and $\tilde{y} = \arg\max_{y \in \mathcal{Y}_{\tilde{t}}} H_{\tilde{t}}(\tilde{x})_y$. Here $\tilde{t}$ and $\tilde{y}$ are the predicted task-id and label for input $\tilde{x}$ respectively, and the subscript y corresponds to the logit for label y. Thus, overall performance depends not only on each task classifier model but also on the accuracy of task-id prediction.

This work employs the setting where the shared parameter θ is frozen after learning the first task [27] and task-specific parameters belonging to previous tasks $\{\theta_j, \phi_j\}_{j<t}$ are also frozen. Although this approach solves forgetting in parameter level, the task-id prediction accuracy (and consequently the overall performance) can degrade because of feature overlap between tasks [28, 19] as representations become shortcut-dominant (Sec 4).

Diffusion Model based Augmentation for CL: The pretrained text-conditioned diffusion model can be used to augment the training data for each incoming task t. Let $\tilde{P}_t$ denote the synthetic data distribution induced by the diffusion model when conditioned on class names associated with task t. During task t, the current task classifier can be trained using a mixture distribution, controlled by mixing ratio λ: $P_t^{\mathrm{mix}} := (1-\lambda)\, P_t + \lambda\, \tilde{P}_t$, by minimizing the cross-entropy loss on samples

$(X, Y) \sim P_t^{\text{mix}}$:

$$\min_{\theta,\theta_t,\phi_t} \mathbb{E}_{(X,Y)\sim P_t^{\text{mix}}}\big[\mathcal{L}_{\text{CE}}(H_t(X), Y)\big], \tag{2}$$

The overall framework can be visualized in Figure 2.

4 Class-Mixed Diffusion Augmentation for Shortcut-Breaking

In parameter-allocation CIL, overall accuracy hinges on *task-id prediction*. A key failure mode is that task-specific feature extractors become *shortcut-dominant*, causing different tasks to induce highly overlapping representations and making task-id prediction fundamentally ambiguous. Motivated by this, we introduce the following formalism:

Notation for Shortcut Learning: Let $S : \mathcal{X} \to \mathcal{S}$ be a map that extracts a candidate shortcut feature (*e.g.,* background) defining the random variable: $S := S(X)$. Similarly, let $R : \mathcal{X} \to \mathcal{R}$ be a map extracting content feature (*e.g.,* object) and defining: $R := R(X)$. Consider a feature extractor map: $f : \mathcal{X} \to \mathcal{Z}$ defining $Z := f(X)$. A task-id predictor is any measurable $g : \mathcal{Z} \to \{1, ...K\}$. The bayes optimal task-id prediction error in then given by $\mathcal{E}^*(Z) := \inf_g \Pr[g(Z) \neq T]$. For each task t, let the task conditional feature distribution of Z be $P_{Z,t} := P(Z \mid T = t)$.

Definition 4.1 (Shortcut-Dominant and Shortcut-Collapse)**.** A learned representation $Z = f(X)$ is ε-*shortcut-dominant* with respect to (S, R) if for task t: $I(Z; R \mid S, T = t) \leq \varepsilon$. This means, conditional on the shortcut S, the representation Z contains at most ε additional information about the content R. We are interested in the *shortcut-dominant* regime where ε is small relative to the scale of the problem: $\varepsilon \ll I(Z; R \mid T = t)$. In the extreme case where Z can be described by the shortcut S alone and is some function of S: $Z = h(S)$, we denote this as shortcut-collapse.

Simple Augmentation Strategy using Diffusion Models (Strategy A): A straightforward method is to use diffusion models to augment the learning of the current task, where diverse samples from the same class are generated. We argue that this augmentation, however do not change the joint dependence structure between shortcut and label within each task, *i.e.,* for each t: $P_{A,t}(S \mid Y) \approx P_t(S \mid Y)$. Strategy A may therefore still produce shortcut-dominant representations. To analyze the resulting task-ID ambiguity, we consider the idealized shortcut-collapse regime $Z_A = h_A(S)$ for a measurable function h_A. We discuss departures from these idealized conditions in Appendix A.2.3.

With this, we introduce the following lemma:

Lemma 4.2. *Let $T \sim$ Unif$(\{1, ..., K\})$ and let $Z_A = f_A(X)$ be the representation from feature extractor learned using strategy A. Assume shortcut-collapse* i.e., *there exist a measurable $h_A : Z_A = h_A(S)$. Assume for a fixed subset of tasks $\mathcal{G} \subseteq \{1, ..., K\}$ with $|\mathcal{G}| = q$ share the same shortcut distribution $P(S \mid T = t) = P(S \mid T = t')\forall t, t' \in \mathcal{G}$. Then, $P(Z_A \mid T = t) = P(Z_A \mid T = t')\forall t, t' \in \mathcal{G}$, and any task-ID predictor g: $\mathcal{Z} \to \{1, ..., K\}$ satisfies $\Pr[g(Z_A) = T \mid T \in \mathcal{G}] \leq 1/q$. Consequently, the Bayes optimal task-ID error obeys:*

$$\mathcal{E}^*(Z_A) := \inf_g \Pr[g(Z) \neq T] \geq \frac{q-1}{K} \tag{3}$$

The detailed proof is provided in Appendix A.2.1.

Remark 4.3 (Shortcut collapse as a fundamental limit for task-id prediction)*.* Lemma 4.2 shows that under shortcut-collapse, task-id prediction is fundamentally limited by the size q of the largest subset of tasks that share an identical shortcut distribution. When $q = K$, all tasks share the same shortcut distribution and $\mathcal{E}^*(Z_A) \geq 1 - \frac{1}{K}$, meaning that task-id prediction degenerates to random guessing among K tasks. As q decreases, the task-id error bound is reduced, motivating our shortcut-breaking augmentation strategy:

Proposed Class Mixing Augmentation Strategy (CMDA): For each task t, we first train a task model using the vanilla diffusion augmentation strategy (Strategy A). CMDA then augments task t with *auxiliary* classes constructed from text tokens that are *not* the class name of any $y \in \mathcal{Y}_t$ but can help break shortcut-collapse. Concretely, we assign each selected token to a new auxiliary label, increasing the number of labels for task t. We introduce one auxiliary class per in-task class. Figure 2 depicts the full pipeline, and Figure 1 illustrates the token selection intuition.

Formally, for each task t and class $y \in \mathcal{Y}_t$, let $\pi_{t,y}$ be a sampling distribution over the diffusion model's text vocabulary (CLIP tokens) that excludes the class-name token(s) of y. CMDA samples m candidate tokens $\mathcal{W}_y = \{W_{y,1}, \ldots, W_{y,m}\} \overset{\text{i.i.d.}}{\sim} \pi_{t,y}$. For each candidate $w \in \mathcal{W}_y$, we generate N_w images conditioned on w, denoted $\{x_{w,i}\}_{i=1}^{N_w}$, and score the candidate using the empirical cross-entropy under the Strategy A classifier:

$$\mathcal{L}_{\mathrm{CE}}(w) \;:=\; -\frac{1}{N_w}\sum_{i=1}^{N_w} \log \sigma_t^A(x_{w,i}, y), \tag{4}$$

where $\sigma_t^A(x, y)$ is the softmax probability assigned to label y by the classifier trained with Strategy A. We select the token with the minimum score:

$$W_y^* \;:=\; \arg\min_{w \in \mathcal{W}_y} \mathcal{L}_{\mathrm{CE}}(w), \tag{5}$$

and treat images generated from W_y^* as an auxiliary class during task-t training. The auxiliary classes and corresponding weights are discarded during inference. During token selection, the frozen diffusion model first generates complete images from discrete vocabulary tokens. CE is then used only to rank the candidate tokens, without optimizing pixels, diffusion latents, text embeddings, or diffusion-model weights. Averaging CE across multiple independent generations per token reduces sensitivity to isolated artifacts. Algorithm 1 in the Appendix A.3 summarizes the overall procedure.

We next analyze minimum-CE token selection and examine how shortcut sharing limits task-id prediction.

Definition 4.4 (Shortcut-Breaking (SB) Token)**.** A token w is Shortcut Breaking Token for task t if using its generations as an auxiliary class reduces shortcut–label dependence: $I_{P_{w,t}}(S;Y) < I_{P_t}(S;Y)$. Here $P_{w,t}$ is the data distribution induced by augmentation with token w for task t. Consequently, let $\mathcal{B}_y \subseteq \mathcal{W}_y$ be the set of shortcut-breaking tokens in the vocabulary for class y.

Lemma 4.5 (Minimum CE loss selects shortcut-breaking tokens)**.** *Fix a task t and class $y \in \mathcal{Y}_t$. Let $\pi_{t,y}$ be the token sampling distribution and $\mathcal{B}_{t,y}$ the set of shortcut-breaking tokens, with $p_y := \pi_{t,y}(\mathcal{B}_{t,y}) > 0$. Let W_y^* denote the token selected by CMDA via minimum empirical cross-entropy loss over m i.i.d. samples from $\pi_{t,y}$. Assume: **(A1)** $\sigma_t^A(x, y) = q_y(S(x))$ almost surely for some measurable q_y; **(A2)** There exist $L_B < L_N$ such that $\sup_{w \in \mathcal{B}_{t,y}} \mathcal{L}(w) \le L_B$ and $\inf_{w \notin \mathcal{B}_{t,y}} \mathcal{L}(w) \ge L_N$, where $\mathcal{L}(w) := \mathbb{E}[-\log \sigma_t^A(X_w, y)]$; **(A3)** $-\log \sigma_t^A(X, y) \in [0, B]$ almost surely. Then, for $\Delta := L_N - L_B > 0$, $\epsilon = \Delta/2$, and with $\delta \;:=\; 2m \exp\!\Big(-\frac{N_w \Delta^2}{2B^2}\Big)$, we have*

$$\Pr\big[W_y^* \in \mathcal{B}_{t,y}\big] \;\ge\; \big(1 - (1 - p_y)^m\big)\,(1 - \delta) \tag{6}$$

We present the proof in Appendix A.2.2. The SB property is a theoretical characterization only and CMDA does not require observing shortcut features or identifying SB tokens directly. Instead, it ranks candidates using empirical CE. The selection bound follows from the positive population-loss gap in A2, i.i.d. token and image sampling, and the bounded-loss assumption in A3. A1 provides the shortcut-based interpretation of the CE score.

Corollary 4.6 (Improvement over random selection)**.** *Under the assumptions of Lemma 4.5, consider the baseline that samples m i.i.d. tokens from $\pi_{t,y}$ and then selects one uniformly at random from the m candidates; denote the selected token by W_y^{rand}. Then $\Pr[W_y^{\mathrm{rand}} \in \mathcal{B}_{t,y}] = p_y$. CMDA outperforms random selection whenever $(1-\delta)\big(1-(1-p_y)^m\big) \;\ge\; p_y$. In particular, if $mp_y \le 1$ and $\delta \le 1 - \frac{2}{m}$ (so $1 - \delta \ge \frac{2}{m}$), then $\Pr[W_y^* \in \mathcal{B}_{t,y}] \ge p_y$.*

The proof is provided in Appendix A.2.5.

Remark 4.7 (Minimum-CE increases the chance of selecting shortcut-breaking tokens). Lemma 4.5 shows that the minimum-CE amplifies the probability that the selected token is shortcut-breaking. Under shortcut-collapse (A1), the classifier logit for label y depends primarily on the shortcut feature $S(x)$; hence, the per-sample loss $\ell(x) = -\log \sigma_t^A(x, y)$ is low on generated samples whose shortcuts are compatible with y. If shortcut-breaking tokens induce shortcut distributions, their population CE loss ($\mathcal{L}(w)$) is smaller than that of non-breaking tokens. Consequently, selecting the token with the smallest empirical CE loss biases the choice toward $\mathcal{B}_{t,y}$, thereby increasing the chance of recovering a shortcut-breaking token.

Now we connect Lemma 4.2 and Lemma 4.5 to show that selecting shortcut-breaking tokens reduces the task-id prediction error:

Definition 4.8 (Largest shortcut-sharing group size)**.** For a representation $Z = f(X)$, define $q(Z) := \max\{|\mathcal{G}| : \mathcal{G} \subseteq \{1, \dots, K\} \text{ s.t. } P(S \mid T = t) = P(S \mid T = t')\ \forall t, t' \in \mathcal{G}\}$ *i.e.,* $q(Z)$ is the size of the largest subset of tasks that share an identical shortcut distribution. For the theoretical analysis below, we fix one largest shortcut-sharing group $\mathcal{G}$ under Strategy A. For each strategy, we evaluate $q(Z)$ using its induced shortcut distributions and restrict the maximum to subsets of this fixed group.

Definition 4.9 (Shortcut-breaking augmentation)**.** Fix task t. Let $P_A(S \mid T = t)$ denote the shortcut distribution induced by Strategy A, and let $P_M(S \mid T = t)$ denote the shortcut distribution induced by a mixed augmentation strategy M. We call M a *shortcut-breaking augmentation* for task t if M augments task t with auxiliary-class data generated from at least one shortcut-breaking token $W_y^* \in \mathcal{B}_{t,y}$, for some $y \in \mathcal{Y}_t$, and changes the shortcut distribution relative to Strategy A: $P_M(S \mid T = t) \neq P_A(S \mid T = t)$.

Theorem 4.10 (Conditional reduction of the shortcut-sharing lower bound)**.** *Fix a task t. Let Z_A and Z_M be the representations learned by Strategy A and CMDA, respectively. Assume Lemma 4.5 holds for each class $y \in \mathcal{Y}_t$, with $0 \leq \delta < 1$, and assume the per-class SB-selection events are mutually independent. Assume further that selecting at least one SB token in task t implies $q(Z_M) < q(Z_A)$. Define $\underline{\mathcal{E}}(Z) := \frac{q(Z)-1}{K}$, which, under conditions of Lemma 4.2, is a lower bound on the Bayes-optimal task-ID error for Z and with $\alpha_t := 1 - \prod_{y \in \mathcal{Y}_t} \big(1 - (1-\delta)\big(1 - (1-p_y)^m\big)\big)$:*

$$\Pr\big[q(Z_M) < q(Z_A)\big] \;\geq\; \alpha_t, \qquad \textit{and} \qquad \Pr[\underline{\mathcal{E}}(Z_M) < \underline{\mathcal{E}}(Z_A)] \geq \alpha_t. \tag{7}$$

Proof sketch. Lemma 4.5 implies that for each class y of task t, CMDA selects an SB token with probability at least $\big(1 - (1-p_y)^m\big)(1-\delta)$. By the assumed independence across classes, the probability that at least one SB token is selected for task t is at least α_t. By the stated group-reduction assumption, this event implies $q(Z_M) < q(Z_A)$. Therefore, $q(Z_M) < q(Z_A)$ holds with probability at least α_t. Finally, Lemma 4.2 lower bounds the task-id error in terms of q; decreasing q strictly decreases this lower bound, implying that $\underline{\mathcal{E}}(Z_M) < \underline{\mathcal{E}}(Z_A)$ with nonzero probability. □

We provide a detailed proof in Appendix A.2.4. The theoretical analysis is conditional on the stated assumptions. Our theory supports that minimum-CE selection is more likely than random to recover a shortcut-breaking token, and links shared shortcuts to task-ID error floor. It does not quantify how much of the accuracy gain on natural benchmarks is attributable to reduced shortcut reliance. We interpret the natural-data analyses as evidence consistent with the proposed mechanism, and discuss bounded departures from the idealized assumptions in Appendix A.2.3.

Remark 4.11 (On Definition 4.9 and why auxiliary classes reduce shortcut sharing). Definition 4.9 formalizes the intended effect of CMDA: once a shortcut-breaking (SB) token is selected and its generations are introduced as an *auxiliary class*, the shortcut S becomes less predictive of the original labels within task t, thereby altering the task-level shortcut distribution and reducing shortcut sharing across tasks. Intuitively, SB tokens are chosen to yield low CE for label y under the Strategy A model, which (under shortcut-dominant scoring) implies that the generated images overlap with the shortcut region associated with y. When these same images are assigned a *new* label (the auxiliary class), they induce *shortcut–label collisions*: similar shortcut realizations now correspond to multiple labels. To minimize training loss, the model cannot rely solely on S and is forced to incorporate additional content information R, which decreases shortcut dominance and reduces the mutual information $I(S;Y)$ within task t. At the distribution level, this corresponds to a shift in the shortcut statistics for the task model relative to Strategy A (e.g., changes in $P(S \mid Y, T = t)$ and hence $P(S \mid T = t)$), which breaks ties between tasks that previously shared identical shortcut distributions.

4.1 Practical Implications

CMDA samples text tokens from the diffusion model vocabulary while excluding the class-name token(s) for label y. In practice, enforcing strict disjointness is non-trivial because many near-synonyms (or morphological variants) of the class name may exist in the vocabulary and are difficult to enumerate. Instead, we operationalize "disjointness" by sampling tokens from *controlled similarity bands* with respect to the class name.

Similarity-thresholded sampling: Rather than enforcing strict token disjointness from the class name (which is difficult due to synonyms), we control candidate selection using a cosine-similarity threshold. Let $\psi(\cdot)$ be the text encoder of the diffusion model and define cosine similarity $\mathrm{sim}(w, y) := \langle \bar{e}(w), \bar{e}_y \rangle$, where $\bar{e}(w)$ is the normalized embedding of token w and $\bar{e}_y$ is the normalized embedding of the class name y (defined below for multi-token names). Given a threshold $\tau \in [-1, 1]$, if $\mathcal{V}$ is the set of token vocabulary, we define the feasible token set as

$$\mathcal{V}_{t,y}(\tau) \;:=\; \big\{ w \in \mathcal{V} :\; \mathrm{sim}(w, y) \leq \tau \big\} \tag{8}$$

and a restricted sampling distribution $\pi^{\tau}_{t,y}$ supported on $\mathcal{V}_{t,y}(\tau)$. CMDA then draws m candidate tokens $\mathcal{W}_y = \{W_{y,1}, \ldots, W_{y,m}\} \sim \pi^{\tau}_{t,y}$, evaluates each candidate using the empirical cross-entropy loss $\mathcal{L}_{\mathrm{CE}}(w)$, and selects $W^*_y = \arg\min_{w \in \mathcal{W}_y} \mathcal{L}_{\mathrm{CE}}(w)$.

The threshold τ controls token–class similarity in text-embedding space, while CE reflects how strongly the Strategy-A classifier associates the generated images with class y. Low CE therefore serves as a classifier-dependent proxy for feature overlap but does not directly measure shortcut overlap or guarantee semantic quality.

Exploring similarity regions: For experimental analysis, we vary τ to probe which portion of the token space yields the most effective shortcut-breaking candidates. Small τ enforces strongly dissimilar tokens (broad exploration but less shortcut overlap), while large τ admits near-synonyms (higher semantic overlap but risk of label ambiguity when treated as auxiliary classes). Varying τ therefore provides a controlled way to quantify how similarity to the class name mediates the trade-off between (i) preserving enough shortcut overlap for effective shortcut–label collisions and (ii) avoiding near-duplicate semantics that can harm the current task boundary.

Relation to the theoretical sampling model: Our theoretical analysis assumes i.i.d. sampling from $\pi_{t,y}$ to obtain clean, nonzero-probability guarantees. The practical procedure above intentionally departs from pure i.i.d. sampling by introducing structured, non-independent candidate construction in order to *target* specific similarity regions and to systematically study their contribution. The resulting τ based sampling is therefore best viewed as an empirical instantiation of CMDA that facilitates controlled ablations over similarity, rather than an exact match to the i.i.d. assumptions used for the theoretical guarantee.

Computation Cost Analysis: Regarding generation time, Strategy A avoids token-selection overhead. CMDA adds a one-time token selection overhead ($O(mN_w)$) per class, but still relies on *vanilla* sampling thereafter. In contrast, gradient-guided generation pays a higher per-image cost for every generated sample. For moderate mN_w (small token-search budget), CMDA can be more efficient than gradient-guided generation with a slight overhead over pure vanilla sampling. The detailed analysis is given in Appendix A.6.

5 Results

Experiment Setup: We use three benchmark datasets: CIFAR100, CUB200, and ImageNet100 (ImageNetSubset), where the following class/task splits are designed for CIL: For CIFAR100: 10 task split (split-CF100) with 10 classes per task. For CUB200, 10 task split (split-CUB) with 20 classes per task. For ImageNet100: 10 task split (split-ImgNet) with 10 classes per task. We use ResNet18 for the CUB200 and ImageNet datasets and ResNet32 for CIFAR100. We use Stable Diffusion V1.5 for the Diffusion Model. Additional details for the classifier and generation are provided in Appendix A.4. For the evaluation metric, we use the Final Accuracy, which is the average accuracy on each task's data evaluated after the model has finished training the final task.

CIL Evaluation: In Table 1, we evaluate our method against EFCIL methods that can be categorized into *feature-based* methods: FeTril [23], PASS [39], and SEED[27]; augmentation-based method: IL2A [38]; *inversion-based* methods: ABD [32], R-DFCIL [7], and SSRE [36]; *diffusion-based* methods: DiffClass [21], FPCIL [14].

Table 1 shows that our method consistently outperforms prior EFCIL approaches on both split-CF100 and split-ImageNet. Compared to the baselines, our approach leverages classifier feedback during generation to explicitly identify and select shortcut-breaking tokens. This allows the synthesized samples to challenge shortcut-dominant representations learned in earlier tasks, leading to more discriminative and transferable features for task-ID prediction. Compared to zero-shot DM baselines

such as FPCIL, which mix classes without explicit control over shortcut reliance, our method benefits from using classifier loss as a signal to filter generations that reduce shortcut overlap across tasks. The benefit of auxiliary-class augmentation compared the contribution of classifier-guided token selection is explained with the ablation studies later in this section.

We use '*' to denote our own implementation. We re-implement DiffClass in our setting for a fair comparison, while retaining its original loss functions. For FPCIL, as assuming access to true future classes is unrealistic, we report the *FPCIL-partial* results from the original paper. For Split-CUB, as it is a fine-grained and challenging benchmark, we start with a pretrained backbone for all methods to ensure a fair comparison (only for Table 1). We evaluate PASS and IL2A on Split-CUB under the same pretrained-backbone setting, with hyperparameters selected through a parameter search. We exclude [35] from the main table as it is intended for a few-shot CL setting. Moreover, generating hard samples is explored in Table 2.

Table 1: Comparison with competitive baselines.

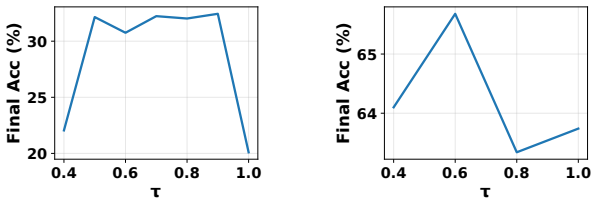
Baseline	Split-CF100	Split-ImgNet	Split-CUB
ABD	34.48	35.66	18.20*
R-DFCIL	42.17	42.70	30.10*
SSRE	29.25	29.28	21.30*
PASS	36.08	16.18	49.73*
FeTril	30.28	27.92	22.30*
SEED	51.42	55.17	23.20*
IL2A	24.49	26.04	38.10*
FPCIL	45.30	57.21	31.80*
DiffClass	53.71*	63.20*	52.08*
Ours	**57.15**$\pm$0.25	**65.51**$\pm$0.22	**56.76**$\pm$0.39

Token Selection Analysis: Table 2 analyzes how classifier-guided token selection and the scope of synthetic samples affect continual learning performance. We sample candidate tokens from the vocabulary and select them using two axes: (i) *similarity control* via a threshold τ in CLIP space, where NEAR corresponds to $\tau = 1$ and FAR uses the best-performing setting with $\tau < 1$; (ii) *classifier-based filtering*, where EASY and HARD select tokens with low vs. high cross-entropy (CE) loss. EASY SAME refer to sampled near tokens that are easy but treated as the same class instead of the auxiliary class, whereas the tokens sampled via RANDOM, NEAR HARD, FAR HARD, NEAR EASY, and FAR EASY are treated as the auxiliary class. STRATEGY A denotes vanilla zero-shot diffusion augmentation without our classifier-informed token filtering. RANDOM samples tokens uniformly without filtering. We use training from scratch for all datasets in Table 2. The standard deviations are provided in the Appendix A.5.2.

Table 2: Impact of generation methods for CIL

Method	Split-CF100	Split-ImgNet	Split-CUB
STRATEGY A	54.35	60.14	19.34
EASY SAME	53.71	63.20	24.59
RANDOM	56.28	64.74	30.53
NEAR HARD	56.68	65.34	28.67
FAR HARD	56.21	65.38	30.83
NEAR EASY	56.08	63.74	20.10
FAR EASY	**57.15**	**65.51**	**32.23**

Classifier-informed token selection consistently improves performance over RANDOM and STRATEGY A, highlighting the benefit of exploiting classifier signals to screen tokens. FAR EASY performs best across all datasets, favoring tokens that are moderately dissimilar to the class name in text-embedding space but whose generations receive low CE for the original class. The HARD variants select higher-CE tokens and achieve lower accuracy, supporting the usefulness of classifier-guided selection. These CE scores reflect the classifier's association with the original class, rather than directly measuring the semantic quality or label correctness of the generated images. Furthermore, EASY SAME also underperforms in most settings when compared to auxiliary class augmentations and always when compared to FAR EASY. This supports the idea of using an auxiliary class instead of generating more samples for the same class. Compared with Strategy A, RANDOM improves CIL accuracy by a good margin where as, FAR EASY provides additional but smaller gains over RANDOM. The larger gain from RANDOM suggests that use of auxiliary classes with RANDOM token selection already has some chance of selecting shortcut-breaking tokens. Our classifier-based method, FAR EASY, helps improve this chance further. This is also consistent with our theoretical analysis following Corollary 4.6.



(a) Split-CUB (b) Split-ImgNet

Figure 3: Helpfulness of different similarity regions.

Which similarity region is most helpful? Fig. 3 studies how the similarity threshold τ affects the performance. We observe a consistent trend: **intermediate similarity** tokens yield the strongest

results, while extremes are suboptimal. In particular, using the **near** region ($\tau = 1$) leads to a clear drop in performance, suggesting that when near-synonymous tokens are treated as an auxiliary class, it hurts learning of current task information (although it may provide shortcut-breaking signals).

Table 7 in Appendix further separates the roles of text similarity and classifier response. At each fixed threshold, low-CE tokens achieve higher accuracy than high-CE tokens, showing that classifier-guided selection provides useful information beyond text similarity. The selected low CE also increases as τ decrease as candidates become more distant in text space. CE therefore serves as a model-dependent proxy for useful feature overlap, rather than a direct measure of shortcut overlap or label correctness.

Overall, these results motivate sampling from *moderately dissimilar* similarity bands (rather than $\tau = 1$) when selecting shortcut-breaking tokens. The qualitative pattern is robust across two datasets, but the exact optimum is dataset-dependent (broader around 0.7–0.9 for CUB, narrower around 0.6 for ImgNet).

Task-ID prediction and connection to shortcut reduction: To better understand where the performance gain comes from, we report both task-ID prediction (TIP) accuracy and task-aware incremental learning (TIL) accuracy in Table 3. Across all benchmarks, CMDA yields substantially larger gains in TIP than in TIL. TIP result for intermediate tasks is given in Appendix A.5.5.

Table 3: TIP and TIL before and after CMDA.

Dataset	TIP		TIL	
	Before	After	Before	After
Split-CF100	55.87	59.25	89.78	90.09
Split-ImgNet	63.18	66.84	92.70	93.38
Split-CUB	26.34	44.19	54.88	58.14

The larger improvement in task-ID prediction is consistent with reduced shortcut reliance. Our theory links shortcut sharing to a task-ID error floor and provides a conditional guarantee for minimum-CE token selection, but does not quantify how much of the accuracy gain on natural benchmarks is attributable to shortcut reduction.

We also perform feature-based analysis using the Hoyer diagnostic, which provides complementary evidence of less concentrated feature use (Appendix A.5.3). The controlled planted-shortcut experiment provides direct evidence that auxiliary labeling reduces reliance on a specific known shortcut, while isolating the labeling component rather than CE-based selection (Appendix A.5.4).

Due to space limitations, we provide additional experiment results in Appendix A.5. Specifically, A.5.2 contains evidence of assumption A2 in Lemma 4.5; A.5.7 includes domain-mismatched evaluation on BloodMnist [5] dataset; A.5.8 contains pretrained-ViT and LoRA based EFCIL evaluations; A.5.9 includes comparison with caption-guided augmentation method [25]; A.5.10 presents few-shot and class imbalanced evaluation; A.5.11 includes analysis of increasing number of auxiliary classes used by our method; and finally, A.5.13 presents computational cost analysis in terms of number of parameters and generation time used by our method.

6 Conclusion

We introduced *Class-Mixed Diffusion Augmentation* (CMDA), a training-free approach that leverages a frozen pretrained diffusion model to generate auxiliary-class training data for EFCIL. We provided theoretical support linking shortcut sharing to task-id prediction error and showed that, under stated assumptions, minimum-CE selection recovers shortcut-breaking tokens with nonzero probability. Across Split-CIFAR100, Split-CUB, and Split-ImageNet, CMDA consistently improves final accuracy over strong EFCIL baselines. Our ablations show that auxiliary-class augmentation contributes most to the observed gain, while classifier-guided selection provides a smaller, consistent improvement over random selection at fixed auxiliary budgets. On natural benchmarks, TIP gains and less concentrated feature use are consistent with reduced shortcut reliance. These benefits come with additional generation and selection costs and dependence on the frozen generator and its vocabulary.

References

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3931–3940, 2020.

[2] Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 10865–10873, 2024.

[3] Reyhane Askari Hemmat, Melissa Hall, Alicia Sun, Candace Ross, Michal Drozdzal, and Adriana Romero-Soriano. Improving geo-diversity of generated images with contextualized vendi score guidance. In *European Conference on Computer Vision*, pages 213–229. Springer, 2024.

[4] Junsong Chen, Shuchen Xue, Yuyang Zhao, Jincheng Yu, Sayak Paul, Junyu Chen, Han Cai, Song Han, and Enze Xie. Sana-sprint: One-step diffusion with continuous-time consistency distillation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16185–16195. IEEE, 2025.

[5] Mohammad Mahdi Derakhshani, Ivona Najdenkoska, Tom van Sonsbeek, Xiantong Zhen, Dwarikanath Mahapatra, Marcel Worring, and Cees GM Snoek. Lifelonger: A benchmark for continual disease classification. In *International conference on medical image computing and computer-assisted intervention*, pages 314–324. Springer, 2022.

[6] Xuefeng Du, Yiyou Sun, Jerry Zhu, and Yixuan Li. Dream the impossible: Outlier imagination with diffusion models. *Advances in Neural Information Processing Systems*, 36:60878–60901, 2023.

[7] Qiankun Gao, Chen Zhao, Bernard Ghanem, and Jian Zhang. R-dfcil: Relation-guided representation learning for data-free class incremental learning. In *European Conference on Computer Vision*, pages 423–439. Springer, 2022.

[8] Rui Gao and Weiwei Liu. Ddgr: Continual learning with deep diffusion-based generative replay. In *International Conference on Machine Learning*, pages 10744–10763. PMLR, 2023.

[9] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost Van De Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36:6582–6595, 2023.

[10] Dipam Goswami, Albin Soutif-Cormerais, Yuyang Liu, Sandesh Kamath, Bart Twardowski, Joost Van De Weijer, et al. Resurrecting old classes with new data for exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28525–28534, 2024.

[11] Patrik O. Hoyer. Non-negative matrix factorization with sparseness constraints. *CoRR*, cs.LG/0408058, 2004.

[12] Khawar Islam and Naveed Akhtar. Context-guided responsible data augmentation with diffusion models. *arXiv preprint arXiv:2503.10687*, 2025.

[13] Khawar Islam, Arif Mahmood, Xin Jin, and Naveed Akhtar. S 2-fracmix: Label-preserving self-saliency mixup augmentation. In *European Conference on Computer Vision*, pages 55–73. Springer, 2026.

[14] Quentin Jodelet, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. Future-proofing class-incremental learning. *Machine Vision and Applications*, 36(1):16, 2025.

[15] Quentin Jodelet, Xin Liu, Yin Jun Phua, and Tsuyoshi Murata. Memory augmented using diffusion model for class-incremental learning. *Image and Vision Computing*, page 105600, 2025.

[16] Gyuhak Kim, Bing Liu, and Zixuan Ke. A multi-head model for continual learning via out-of-distribution replay. In *Conference on Lifelong Learning Agents*, pages 548–563. PMLR, 2022.

[17] Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical study on solving continual learning. *Advances in neural information processing systems*, 35:5065–5079, 2022.

[18] Jinglin Liang, Jin Zhong, Hanlin Gu, Zhongqi Lu, Xingxing Tang, Gang Dai, Shuangping Huang, Lixin Fan, and Qiang Yang. Diffusion-driven data replay: A novel approach to combat forgetting in federated class continual learning. In *European Conference on Computer Vision*, pages 303–319. Springer, 2024.

[19] Haowei Lin, Yijia Shao, Weinan Qian, Ningxin Pan, Yiduo Guo, and Bing Liu. Class incremental learning via likelihood ratio based task prediction. In *The Twelfth International Conference on Learning Representations*, 2024.

[20] Xuan Liu and Xiaobin Chang. Lora subtraction for drift-resistant space in exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15308–15318, 2025.

[21] Zichong Meng, Jie Zhang, Changdi Yang, Zheng Zhan, Pu Zhao, and Yanzhi Wang. Diffclass: Diffusion-based class incremental learning. In *European Conference on Computer Vision*, pages 142–159. Springer, 2024.

[22] Pablo Pernias, Dominic Rampas, Mats L Richter, Christopher Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *International Conference on Learning Representations*, volume 2024, pages 25097–25109, 2024.

[23] Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. Fetril: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3911–3920, 2023.

[24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[25] Aniket Roy, Anshul Shah, Ketul Shah, Anirban Roy, and Rama Chellappa. Cap2aug: Caption guided image data augmentation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 9125–9135. IEEE, 2025.

[26] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[27] Grzegorz Rypeść, Sebastian Cygert, Valeriya Khan, Tomasz Trzciński, Bartosz Zieliński, and Bartłomiej Twardowski. Divide and not forget: Ensemble of selectively trained experts in continual learning. *arXiv preprint arXiv:2401.10191*, 2024.

[28] Grzegorz Rypeść, Sebastian Cygert, Tomasz Trzciński, and Bartłomiej Twardowski. Task-recency bias strikes back: Adapting covariances in exemplar-free class incremental learning. *Advances in Neural Information Processing Systems*, 37:63268–63289, 2024.

[29] Seyedmorteza Sadat, Jakob Buhmann, Derek Bradley, Otmar Hilliges, and Romann M Weber. Cads: Unleashing the diversity of diffusion models through condition-annealed sampling. *arXiv preprint arXiv:2310.17347*, 2023.

[30] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pages 4548–4557. PMLR, 2018.

[31] Paweł Skierś and Kamil Deja. Joint diffusion models in continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4380–4390, 2025.

[32] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9374–9384, 2021.

[33] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11909–11919, 2023.

[34] Soobin Um, Beomsu Kim, and Jong Chul Ye. Boost-and-skip: A simple guidance-free diffusion for minority generation. *arXiv preprint arXiv:2502.06516*, 2025.

[35] Ruitao Wu, Yifan Zhao, Guangyao Chen, and Jia Li. Diffusion-classifier synergy: Reward-aligned learning via mutual boosting loop for fscil. *arXiv preprint arXiv:2510.03608*, 2025.

[36] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8715–8724, 2020.

[37] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23219–23230, 2024.

[38] Fei Zhu, Zhen Cheng, Xu-yao Zhang, and Cheng-lin Liu. Class-incremental learning via dual augmentation. *Advances in neural information processing systems*, 34:14306–14318, 2021.

[39] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021.

A Appendix

The Appendix is organized as follows: A.1 contains the list of symbols; A.2 contains the proofs; A.3 contains the algorithm; A.4 contains additional implementation details; A.5 contains additional experiment results; A.6 contains computational cost analysis.

A.1 Table of symbols and notation.

Table 4 presents the list of symbols and notations used for our method.

Table 4: Table of symbols and notation.

Symbol	Description
Latent diffusion / generation	
$\mathrm{Enc}, \mathrm{Dec}$	VAE encoder/decoder for latent diffusion; $z_0 = \mathrm{Enc}(x)$, $\tilde{x} = \mathrm{Dec}(z_0)$.
z_i, $\tilde{n}$, $z_{\tilde{n}}$	Reverse diffusion latents and horizon; initialize $z_{\tilde{n}} \sim \mathcal{N}(0, \mathbf{I})$ and iterate $i = \tilde{n}, \ldots, 0$.
x_p	Text prompt (e.g., $\mathrm{concat}(\mathrm{desc}, \mathrm{name})$) conditioning generation.
$\hat{\epsilon}_\beta(\cdot)$, β, $\mathrm{Step}(\cdot)$	U-Net noise prediction with parameters β and a sampler step producing $z_{i-1} = \mathrm{Step}(z_i, i, \hat{\epsilon}_\beta(\cdot))$.
Continual learning setup	
K, T, t	Number of tasks; task-id random variable $T \in \{1, \ldots, K\}$ and task index t.
$P_t(X, Y)$, X, Y, $\mathcal{X}_t$, $\mathcal{Y}_t$	Task-t data distribution and associated image/label spaces.
θ, θ_t, ϕ_t	Shared parameters θ (frozen after task 1) and task-specific parameters (θ_t, ϕ_t).
F_t, H_t	Feature extractor $F_t(X) = F(\theta, \theta_t, X)$ and classifier $H_t(X) = H(F_t(X), \phi_t)$ (logits over $\mathcal{Y}_t$).
$g(\tilde{x})$, $\tilde{t}$, $\tilde{y}$	Maximum-logit routing and predictions: $\tilde{t} = g(\tilde{x})$, $\tilde{y} = \arg\max_{y \in \mathcal{Y}_{\tilde{t}}} H_{\tilde{t}}(\tilde{x})_y$.
Diffusion augmentation mixtures	
$\tilde{P}_t$, λ, P_t^{mix}	Synthetic distribution for task t, mixing ratio λ, and $P_t^{\mathrm{mix}} = (1-\lambda)P_t + \lambda\tilde{P}_t$.
Shortcut/content and task-id limits	
S, R	Shortcut/content features extracted from X (e.g., background vs. object).
f, Z	Representation map $f : \mathcal{X} \to \mathcal{Z}$ and representation $Z = f(X)$.
$I(\cdot;\cdot \mid \cdot)$, ε	Conditional mutual information; ε-shortcut-dominance uses $I(Z; R \mid S, T = t) \le \varepsilon$.
$\mathcal{E}^*(Z)$	Bayes-optimal task-id error: $\mathcal{E}^*(Z) = \inf_g \Pr[g(Z) \neq T]$.
$\mathcal{G}$, q, $q(Z)$	Shortcut-sharing task subset $\mathcal{G}$ with $q = \lvert\mathcal{G}\rvert$; $q(Z)$ is the largest such group size.
Z_A, Z_M	Representations under Strategy A (vanilla) and CMDA/Strategy M; shortcut-collapse modeled as $Z_A = h_A(S)$.
CMDA token selection and guarantee	
$\pi_{t,y}$, m, $\mathcal{W}_y$	Token sampling distribution (excluding class-name), candidates per class, and sampled set $\mathcal{W}_y = \{W_{y,1}, \ldots, W_{y,m}\}$.
N_w, $x_{w,i}$	Images generated to score token w and the i-th generated image conditioned on w.
σ_t^A, $\mathcal{L}_{\mathrm{CE}}(w)$, W_y^*	Strategy-A softmax; empirical CE score; selected token $W_y^* = \arg\min_{w \in \mathcal{W}_y} \mathcal{L}_{\mathrm{CE}}(w)$.
$\mathcal{B}_{t,y}$, p_y	Shortcut-breaking token set and its mass $p_y = \pi_{t,y}(\mathcal{B}_{t,y})$.
Δ, B, δ, α_t	Loss gap $\Delta = L_N - L_B$, bounded loss range $[0, B]$, Hoeffding-bound term δ, and success probability bound α_t.
ψ, $\mathrm{sim}(w, y)$, τ, $\mathcal{V}_{t,y}(\tau)$, $\pi^{\tau}_{t,y}$	Text encoder ψ, cosine similarity, threshold τ, feasible tokens $\mathcal{V}_{t,y}(\tau) = \{w : \mathrm{sim}(w, y) \le \tau\}$, and restricted sampler.

A.2 Proofs

A.2.1 Proof for Lemma 4.2

Proof. Let $T \sim \mathrm{Unif}(\{1, \ldots, K\})$. Let $S = S(X)$ denote the shortcut random variable and $Z_A = f_A(X)$ the learned representation under Strategy A. By the shortcut-collapse assumption, there exists a measurable function $h_A : \mathcal{S} \to \mathcal{Z}$ such that

$$Z_A = h_A(S) \qquad \text{a.s.} \tag{9}$$

Step 1: Shared shortcut distributions imply shared representation distributions within $\mathcal{G}$. Fix any measurable set $A \subseteq \mathcal{Z}$. For each task t, define

$$P_{Z_A,t}(A) := \Pr(Z_A \in A \mid T = t).$$

Using (9) and measurability of h_A, we obtain

$$\begin{aligned} P_{Z_A,t}(A) &= \Pr(h_A(S) \in A \mid T = t) \\ &= \Pr(S \in h_A^{-1}(A) \mid T = t). \end{aligned} \tag{10}$$

Now let $t, t' \in \mathcal{G}$. By assumption, tasks in $\mathcal{G}$ share the same shortcut distribution:

$$P(S \mid T = t) = P(S \mid T = t').$$

Applying this equality to the set $h_A^{-1}(A) \subseteq \mathcal{S}$ and using (10) yields

$$P_{Z_A,t}(A) = \Pr(S \in h_A^{-1}(A) \mid T = t) = \Pr(S \in h_A^{-1}(A) \mid T = t') = P_{Z_A,t'}(A). \tag{11}$$

Since this holds for all measurable $A \subseteq \mathcal{Z}$, we conclude that

$$P(Z_A \mid T = t) = P(Z_A \mid T = t') \qquad \forall\, t, t' \in \mathcal{G}. \tag{12}$$

Step 2: No task-id predictor can exceed $1/q$ accuracy on $\mathcal{G}$. Let $g : \mathcal{Z} \to \{1, \ldots, K\}$ be any measurable task-id predictor. For each $t \in \mathcal{G}$, define the decision region

$$A_t := \{z \in \mathcal{Z} : g(z) = t\}.$$

These sets are pairwise disjoint and satisfy $\bigcup_{t\in\mathcal{G}} A_t \subseteq \mathcal{Z}$.

Conditioning on $T \in \mathcal{G}$, we can write

$$\begin{aligned}\Pr\big(g(Z_A) = T \mid T \in \mathcal{G}\big) &= \sum_{t\in\mathcal{G}} \Pr\big(T = t \mid T \in \mathcal{G}\big)\ \Pr\big(g(Z_A) = t \mid T = t\big) \\ &= \sum_{t\in\mathcal{G}} \Pr\big(T = t \mid T \in \mathcal{G}\big)\ \Pr\big(Z_A \in A_t \mid T = t\big). \end{aligned} \tag{13}$$

Because T is uniform on $\{1, \ldots, K\}$ and $|\mathcal{G}| = q$, we have

$$\Pr(T = t \mid T \in \mathcal{G}) = \frac{1}{q} \qquad \forall\, t \in \mathcal{G}. \tag{14}$$

Moreover, by (12), the conditional distribution of Z_A is the same for all $t \in \mathcal{G}$. Let Q denote this common distribution, i.e.,

$$Q(\cdot) := P(Z_A \in \cdot \mid T = t) \qquad \text{for any } t \in \mathcal{G}.$$

Then $\Pr(Z_A \in A_t \mid T = t) = Q(A_t)$ for all $t \in \mathcal{G}$. Substituting into (13) and using (14) gives

$$\Pr\big(g(Z_A) = T \mid T \in \mathcal{G}\big) = \sum_{t\in\mathcal{G}} \frac{1}{q}\, Q(A_t) = \frac{1}{q} \sum_{t\in\mathcal{G}} Q(A_t). \tag{15}$$

Since $\{A_t\}_{t\in\mathcal{G}}$ are disjoint and Q is a probability measure,

$$\sum_{t\in\mathcal{G}} Q(A_t) \le 1,$$

and therefore

$$\Pr\big(g(Z_A) = T \mid T \in \mathcal{G}\big) \le \frac{1}{q}. \tag{16}$$

Step 3: Bayes optimal task-id error lower bound. From (16),

$$\Pr\big(g(Z_A) \neq T \mid T \in \mathcal{G}\big) = 1 - \Pr\big(g(Z_A) = T \mid T \in \mathcal{G}\big) \ge 1 - \frac{1}{q}.$$

Thus, for any g,

$$\begin{aligned}\Pr\big(g(Z_A) \neq T\big) &\ge \Pr(T \in \mathcal{G})\ \Pr\big(g(Z_A) \neq T \mid T \in \mathcal{G}\big) \\ &\ge \Pr(T \in \mathcal{G}) \left(1 - \frac{1}{q}\right). \end{aligned} \tag{17}$$

Finally, since T is uniform and $|\mathcal{G}| = q$, $\Pr(T \in \mathcal{G}) = q/K$. Plugging this into (17) yields

$$\Pr\big(g(Z_A) \neq T\big) \ge \frac{q}{K} \left(1 - \frac{1}{q}\right) = \frac{q-1}{K}.$$

Taking the infimum over all measurable predictors g proves

$$\mathcal{E}^*(Z_A) := \inf_g \Pr\big(g(Z_A) \neq T\big) \ \ge\ \frac{q-1}{K}.$$

□

A.2.2 Proof for Lemma 4.5

Assumptions for Lemma 4.5. Fix a task t and a class $y \in \mathcal{Y}_t$. For any token w, let $P_{w,t}$ denote the distribution over generated images induced by conditioning the diffusion model on w (and the prompt template used for task t). Let $\sigma_t^A(x, y)$ denote the Strategy A model's predicted probability for label y on input x.

Assumption A.1 (Shortcut-dominant scoring under Strategy A)**.** There exists a measurable function $q_y : \mathcal{S} \to [0, 1]$ such that, for $X \sim P_{w,t}$ (and also for $X \sim P_t$),

$$\sigma_t^A(X, y) = q_y(S(X)) \quad \text{a.s.}$$

That is, the Strategy A confidence for class y depends on the input only through the shortcut feature $S(X)$.

Assumption A.2 (Population loss gap between shortcut-breaking and non-breaking tokens)**.** Define the population cross-entropy loss of a token w (w.r.t. class y) as

$$\mathcal{L}(w) \;:=\; \mathbb{E}_{X\sim P_{w,t}}\big[-\log \sigma_t^A(X, y)\big].$$

Let $\mathcal{B}_{t,y}$ denote the set of shortcut-breaking tokens for (t, y). There exist constants $L_B < L_N$ such that

$$\sup_{w\in\mathcal{B}_{t,y}} \mathcal{L}(w) \;\le\; L_B, \qquad \inf_{w\notin\mathcal{B}_{t,y}} \mathcal{L}(w) \;\ge\; L_N.$$

We denote the resulting margin by $\Delta := L_N - L_B > 0$.

Assumption A.3 (Concentration of empirical CE loss)**.** For each candidate token w, we generate N i.i.d. images $X_{w,1}, \ldots, X_{w,N} \sim P_{w,t}$ and compute

$$\widehat{\mathcal{L}}_{\mathrm{CE}}(w) \;:=\; \frac{1}{N}\sum_{i=1}^{N} \ell(X_{w,i}), \qquad \ell(x) := -\log \sigma_t^A(x, y).$$

Assume the per-sample loss is bounded almost surely: $\ell(X) \in [0, B]$ for some finite $B > 0$.

Proof. Fix t and $y \in \mathcal{Y}_t$. Let $\mathcal{B} := \mathcal{B}_{t,y}$ and $p_y := \pi_{t,y}(\mathcal{B}) > 0$. Sample $W_{y,1}, \ldots, W_{y,m} \overset{\text{i.i.d.}}{\sim} \pi_{t,y}$ and let $\mathcal{W}_y = \{W_{y,1}, \ldots, W_{y,m}\}$. For each $w \in \mathcal{W}_y$, compute the empirical CE loss $\widehat{\mathcal{L}}_{\mathrm{CE}}(w)$ on N i.i.d. generated samples, and select $W_y^* = \arg\min_{w\in\mathcal{W}_y} \widehat{\mathcal{L}}_{\mathrm{CE}}(w)$.

Step 1 (probability of sampling an SB token). Since each draw lands in $\mathcal{B}$ with probability p_y,

$$\Pr(\mathcal{W}_y \cap \mathcal{B} = \emptyset) = (1 - p_y)^m, \qquad \Pr(\mathcal{W}_y \cap \mathcal{B} \neq \emptyset) = 1 - (1 - p_y)^m.$$

Step 2 (population CE depends on shortcut distribution). By (A1), $\sigma_t^A(X, y) = q_y(S(X))$ a.s., hence for any token w,

$$\mathcal{L}(w) := \mathbb{E}_{X\sim P_{w,t}}\big[-\log \sigma_t^A(X, y)\big] = \mathbb{E}_{X\sim P_{w,t}}\big[-\log q_y(S(X))\big].$$

Thus $\mathcal{L}(w)$ is determined by the shortcut marginal induced by $P_{w,t}$.

Step 3 (Empirical CE concentrates around population CE). Fix a candidate token $w \in \mathcal{W}_y$. Let $X_{w,1}, \ldots, X_{w,N} \overset{\text{i.i.d.}}{\sim} P_{w,t}$ be the N synthetic images generated conditioned on w. Define the per-sample loss random variable

$$U_{w,i} \;:=\; \ell(X_{w,i}) \quad \text{where} \quad \ell(x) := -\log \sigma_t^A(x, y).$$

Then the empirical CE loss used in the algorithm is the sample average

$$\widehat{\mathcal{L}}_{\mathrm{CE}}(w) \;:=\; \frac{1}{N}\sum_{i=1}^{N} U_{w,i}.$$

The corresponding population (expected) loss is

$$\mathcal{L}(w) \;:=\; \mathbb{E}_{X\sim P_{w,t}}\big[\ell(X)\big] \;=\; \mathbb{E}[U_{w,1}],$$

and by linearity of expectation,

$$\mathbb{E}\big[\widehat{\mathcal{L}}_{\mathrm{CE}}(w)\big] \;=\; \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} U_{w,i}\right] \;=\; \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[U_{w,i}] \;=\; \mathcal{L}(w),$$

since the $U_{w,i}$ are i.i.d.

Assume $U_{w,i} \in [0, B]$ almost surely (e.g., by clipping probabilities so that $\sigma_t^A(x, y) \geq e^{-B}$). Then Hoeffding's inequality for bounded i.i.d. variables yields, for any $\epsilon > 0$,

$$\Pr\Big(\Big|\widehat{\mathcal{L}}_{\mathrm{CE}}(w) - \mathcal{L}(w)\Big| \geq \epsilon\Big) \leq 2\exp\left(-\frac{2N\epsilon^2}{B^2}\right).$$

Crucially, the factor N appears in the exponent because the empirical loss is an average over N i.i.d. terms.

Step 4 (uniform concentration of empirical CE over the m candidates). Let $\ell(x) := -\log \sigma_t^A(x, y) \in [0, B]$ as in (A3), and $\widehat{\mathcal{L}}_{\mathrm{CE}}(w) = \frac{1}{N}\sum_{i=1}^{N} \ell(X_{w,i})$ with $X_{w,i} \overset{\text{i.i.d.}}{\sim} P_{w,t}$. Hoeffding's inequality yields, for any $\epsilon > 0$,

$$\Pr\left(|\widehat{\mathcal{L}}_{\mathrm{CE}}(w) - \mathcal{L}(w)| \geq \epsilon\right) \leq 2\exp\left(-\frac{2N\epsilon^2}{B^2}\right).$$

Applying a union bound over the m sampled candidates gives

$$\Pr\Big(\max_{w\in\mathcal{W}_y} |\widehat{\mathcal{L}}_{\mathrm{CE}}(w) - \mathcal{L}(w)| \geq \epsilon\Big) \leq 2m\exp\left(-\frac{2N\epsilon^2}{B^2}\right).$$

Set $\epsilon = \Delta/2$ and define

$$\delta := 2m\exp\left(-\frac{N\Delta^2}{2B^2}\right).$$

Then with probability at least $1 - \delta$, simultaneously for all $w \in \mathcal{W}_y$,

$$|\widehat{\mathcal{L}}_{\mathrm{CE}}(w) - \mathcal{L}(w)| \leq \Delta/2. \tag{18}$$

Step 5 (if an SB token is present, the empirical minimizer selects SB with prob. $\geq 1 - \delta$). Condition on the event $\mathcal{W}_y \cap \mathcal{B} \neq \emptyset$, and pick any $w_B \in \mathcal{W}_y \cap \mathcal{B}$. For any non-SB candidate $w_N \in \mathcal{W}_y \setminus \mathcal{B}$, by (A2) we have $\mathcal{L}(w_B) \leq L_B$ and $\mathcal{L}(w_N) \geq L_N = L_B + \Delta$. On the concentration event (18),

$$\widehat{\mathcal{L}}_{\mathrm{CE}}(w_B) \leq \mathcal{L}(w_B) + \Delta/2 \leq L_B + \Delta/2,$$

and

$$\widehat{\mathcal{L}}_{\mathrm{CE}}(w_N) \geq \mathcal{L}(w_N) - \Delta/2 \geq (L_B + \Delta) - \Delta/2 = L_B + \Delta/2.$$

Hence $\widehat{\mathcal{L}}_{\mathrm{CE}}(w_B) \leq \widehat{\mathcal{L}}_{\mathrm{CE}}(w_N)$ for all $w_N \notin \mathcal{B}$, so any empirical minimizer W_y^* must satisfy $W_y^* \in \mathcal{B}$. Therefore,

$$\Pr\big(W_y^* \in \mathcal{B} \,\big|\, \mathcal{W}_y \cap \mathcal{B} \neq \emptyset\big) \geq 1 - \delta.$$

This establishes the conditional selection guarantee.

Step 6 (combine Steps 1 and 5). Finally,

$$\Pr(W_y^* \in \mathcal{B}) \geq \Pr(\mathcal{W}_y \cap \mathcal{B} \neq \emptyset)\,\Pr\big(W_y^* \in \mathcal{B} \,\big|\, \mathcal{W}_y \cap \mathcal{B} \neq \emptyset\big) \geq \big(1 - (1 - p_y)^m\big)\,(1 - \delta),$$

which is strictly positive for $p_y > 0$ and $m \geq 1$. □

A.2.3 Sensitivity to Approximate Assumptions

In the above section, we assumed idealized conditions for shortcut-collapse. Here we consider how the guarantees change under bounded departures from these conditions.

Perturbations of shortcut-dependent scoring. Fix a task t and class y. Let $\mathcal{L}_0(w)$ denote a shortcut-only reference population loss (expected CE loss under an idealized classifier whose prediction depends only on the shortcut S) satisfying the separation condition in A2 with gap $\Delta_0 > 0$. Suppose the actual population loss satisfies

$$|\mathcal{L}(w) - \mathcal{L}_0(w)| \le \eta$$

for every token in the support of $\pi_{t,y}$. Then, for any shortcut-breaking token w_B and non-breaking token w_N in this support,

$$\mathcal{L}(w_N) - \mathcal{L}(w_B) \ge \mathcal{L}_0(w_N) - \mathcal{L}_0(w_B) - 2\eta \ge \Delta_0 - 2\eta.$$

Thus, a positive separation is retained whenever

$$\Delta_{\text{eff}} := \Delta_0 - 2\eta > 0.$$

Retaining the same i.i.d. candidate sampling, i.i.d. generations per candidate, and bounded actual CE loss in $[0, B]$, the same concentration argument applies with Δ_{eff} replacing Δ in Lemma 4.5. Using this, we can define the corresponding deviation-probability bound as:

$$\delta_{\text{eff}} := \min\left\{1,\, 2m \exp\left(-\frac{N_w \Delta_{\text{eff}}^2}{2B^2}\right)\right\}.$$

Conditioned on any sampled candidate set, Hoeffding's inequality and a union bound ensure that all empirical losses deviate from their population values by less than $\Delta_{\text{eff}}/2$ with probability at least $1 - \delta_{\text{eff}}$. With this, every sampled shortcut-breaking token has strictly smaller empirical loss than every sampled non-breaking token. Consequently, the probability of selecting a shortcut-breaking token is bounded by:

$$\Pr\left[W_y^* \in \mathcal{B}_{t,y}\right] \ge \left(1 - (1 - p_y)^m\right)(1 - \delta_{\text{eff}}).$$

The bound becomes weaker as the effective gap (Δ_{eff}) narrows. If $\Delta_{\text{eff}} \le 0$, this perturbation argument no longer ensures separation. The selection guarantee follows from a positive population-loss gap (A2) and the sampling and concentration assumptions. Exact A1 is not needed for the bound; its role is to connect CE scores to shortcut features.

Approximate shortcut collapse. Retain the uniform prior over K tasks and consider a group $\mathcal{G}$ of q tasks. Let $P_t = P(Z \in \cdot \mid T = t)$ (entire conditional distribution) and let there is a common shortcut-only reference distribution Q such that

$$d(P_t, Q) \le \gamma \qquad \text{for all } t \in \mathcal{G},$$

where $d(P, Q) := \sup_A |P(A) - Q(A)|$.

For any measurable task-ID predictor g, let $A_t = \{z : g(z) = t\}$. These decision regions are disjoint, so

$$\begin{aligned}
\Pr[g(Z) = T \mid T \in \mathcal{G}] &= \frac{1}{q} \sum_{t \in \mathcal{G}} P_t(A_t) \\
&\le \frac{1}{q} \sum_{t \in \mathcal{G}} \big(Q(A_t) + \gamma\big) \\
&\le \frac{1}{q} + \gamma.
\end{aligned}$$

The accuracy bound is capped at 1. Hence the Bayes-optimal task-ID error satisfies

$$\mathcal{E}^*(Z) \ge \frac{q}{K} \max\left\{0,\, 1 - \frac{1}{q} - \gamma\right\}.$$

When $\gamma = 0$, this recovers the lower bound $(q-1)/K$; positive γ weakens it additively.

A.2.4 Proof for Theorem4.10

Proof. We prove the two claims in Theorem 4.10: (i) $q(Z_M) < q(Z_A)$ with nonzero probability, and (ii) this implies $\underline{\mathcal{E}}(Z_M) < \underline{\mathcal{E}}(Z_A)$ with nonzero probability.

Step 1: A per-class success event. Fix a task t and a class $y \in \mathcal{Y}_t$. Let $\mathcal{B}_{t,y}$ be the set of SB tokens and $p_y := \pi_{t,y}(\mathcal{B}_{t,y}) > 0$. Let $\mathcal{W}_y = \{W_{y,1}, \ldots, W_{y,m}\}$ be the m i.i.d. samples from $\pi_{t,y}$, and let $W_y^* := \arg\min_{w \in \mathcal{W}_y} \widehat{\mathcal{L}}_{\mathrm{CE}}(w)$ be the minimizer of empirical CE over N i.i.d. generated samples per candidate token (as in Lemma 4.5). Define the event

$$E_{t,y} \;:=\; \{W_y^* \in \mathcal{B}_{t,y}\}.$$

By Lemma 4.5, under Assumptions (A1)–(A3),

$$\Pr(E_{t,y}) \;\geq\; \Big(1 - (1-p_y)^m\Big)\,(1-\delta_y), \tag{19}$$

where $\Delta := L_N - L_B > 0$, $\epsilon = \Delta/2$, and (with N samples per token and loss bound B)

$$\delta_y \;:=\; 2m \exp\!\left(-\frac{2N\epsilon^2}{B^2}\right) \;=\; 2m \exp\!\left(-\frac{N\Delta^2}{2B^2}\right).$$

Let $\delta < 1$ be a common upper bound satisfying $\delta_y \leq \delta$ for every $y \in \mathcal{Y}_t$.

Step 2: A per-task success event. Define the per-task event that CMDA selects *at least one* SB token for task t,

$$E_t \;:=\; \bigcup_{y \in \mathcal{Y}_t} E_{t,y}.$$

By the assumed mutual independence of the per-class selection events and the bounds from Step 1,

$$\begin{aligned}
\Pr(E_t) &= 1 - \prod_{y \in \mathcal{Y}_t} \big(1 - \Pr(E_{t,y})\big) \\
&\geq 1 - \prod_{y \in \mathcal{Y}_t} \Big(1 - \big(1 - (1-p_y)^m\big)(1-\delta_y)\Big) \\
&\geq 1 - \prod_{y \in \mathcal{Y}_t} \Big(1 - \big(1 - (1-p_y)^m\big)(1-\delta)\Big) \\
&= \alpha_t.
\end{aligned} \tag{20}$$

Since $p_y > 0$ for at least one $y \in \mathcal{Y}_t$ and $\delta < 1$ for sufficiently large N, we have $\alpha_t > 0$.

Step 3: From E_t to a decrease in shortcut sharing. Let $P_A(S \mid T = \cdot)$ denote the task-conditional shortcut distributions under Strategy A, and $P_M(S \mid T = \cdot)$ those under CMDA. Assume that under Strategy A there exists a largest shortcut-sharing group $\mathcal{G}_A \subseteq \{1, \ldots, K\}$ (the same fixed group $\mathcal{G}$ selected in Definition 4.8) of size

$$q(Z_A) = |\mathcal{G}_A|, \qquad \text{such that} \qquad P_A(S \mid T = t) = P_A(S \mid T = t') \;\; \forall t, t' \in \mathcal{G}_A.$$

For the task t fixed in the theorem, event E_t means that CMDA selects at least one SB token for that task. By the theorem's group-reduction assumption, this implies $q(Z_M) < q(Z_A)$. Since q is integer-valued,

$$q(Z_M) \leq q(Z_A) - 1 \qquad \text{on event } E_t.$$

Therefore,

$$\Pr\big(q(Z_M) < q(Z_A)\big) \;\geq\; \Pr(E_t) \;\geq\; \alpha_t \;>\; 0, \tag{21}$$

(Here we use the law of probability: $P(A) = P(A|E)P(E) + P(A|E^c)P(E^c) \geq P(A|E)P(E) \geq P(E)$ as $P(A|E) = 1$)

which proves the first claim.

Step 4: Reduction of the shortcut-sharing lower bound. Recall that

$$\underline{\mathcal{E}}(Z) := \frac{q(Z) - 1}{K}.$$

Under the conditions of Lemma 4.2, the Bayes-optimal task-ID error satisfies $\mathcal{E}^*(Z) \geq \underline{\mathcal{E}}(Z)$.

On the event $q(Z_M) \leq q(Z_A) - 1$, we have

$$\underline{\mathcal{E}}(Z_M) = \frac{q(Z_M) - 1}{K} \leq \frac{q(Z_A) - 2}{K} < \frac{q(Z_A) - 1}{K} = \underline{\mathcal{E}}(Z_A).$$

Thus, a decrease in q strictly decreases the shortcut-sharing lower-bound expression. Combining this with (21) yields

$$\Pr[\underline{\mathcal{E}}(Z_M) < \underline{\mathcal{E}}(Z_A)] = \Pr[q(Z_M) < q(Z_A)] \geq \alpha_t > 0.$$

This completes the proof.

□

A.2.5 Proof for Corollary 4.6

Proof. Fix task t and class $y \in \mathcal{Y}_t$ and write $\mathcal{B} := \mathcal{B}_{t,y}$ and $p := p_y = \pi_{t,y}(\mathcal{B})$ for brevity.

Step 1: Random-from-m baseline has success probability p. Let $W_1, \ldots, W_m \overset{\text{i.i.d.}}{\sim} \pi_{t,y}$ and let $J \sim \mathrm{Unif}(\{1, \ldots, m\})$ be independent of $\{W_i\}_{i=1}^m$. Define the baseline selection $W^{\mathrm{rand}} := W_J$. Then, by the law of total probability,

$$\begin{aligned} \Pr\big[W^{\mathrm{rand}} \in \mathcal{B}\big] &= \sum_{j=1}^{m} \Pr[J = j] \Pr[W_j \in \mathcal{B}] \\ &= \sum_{j=1}^{m} \frac{1}{m} p \;=\; p. \end{aligned} \tag{22}$$

Step 2: CMDA lower bound (from Lemma 4.5). Lemma 4.5 yields

$$\Pr\big[W^* \in \mathcal{B}\big] \;\geq\; (1 - \delta)\Big(1 - (1 - p)^m\Big), \tag{23}$$

where $\delta := 2m \exp\Big(-\frac{N_w \Delta^2}{2B^2}\Big)$ and $\Delta = L_N - L_B > 0$.

Step 3: Condition for improvement over random. Comparing Step 1 and Step 2, CMDA is guaranteed to outperform the random baseline whenever

$$(1 - \delta)\Big(1 - (1 - p)^m\Big) \;\geq\; p, \tag{24}$$

which is exactly the stated condition.

Step 4: A simple sufficient condition. Assume $mp \leq 1$. Using the elementary inequality

$$1 - (1 - p)^m \;\geq\; 1 - e^{-mp}, \tag{25}$$

(which follows from $1 - x \leq e^{-x}$ for $x \in [0, 1]$), and the bound

$$1 - e^{-u} \;\geq\; \frac{u}{2} \qquad \text{for all } u \in [0, 1], \tag{26}$$

we obtain

$$1 - (1 - p)^m \;\geq\; 1 - e^{-mp} \;\geq\; \frac{mp}{2}. \tag{27}$$

Therefore,

$$\Pr\big[W^* \in \mathcal{B}\big] \;\geq\; (1 - \delta)\frac{mp}{2}. \tag{28}$$

If, in addition, $\delta \leq 1 - \frac{2}{m}$ (equivalently $1 - \delta \geq \frac{2}{m}$), then

$$\Pr\big[W^* \in \mathcal{B}\big] \;\geq\; \frac{2}{m} \cdot \frac{mp}{2} \;=\; p \;= \Pr\big[W^{\mathrm{rand}} \in \mathcal{B}\big], \tag{29}$$

so CMDA is guaranteed to be at least as good as random selection under these sufficient conditions.

□

Algorithm 1 Class-Mixed Diffusion Augmentation (CMDA) for Shortcut-Breaking

Input: tasks $\{1, \ldots, K\}$ with data $\{\mathcal{D}_t\}$; diffusion model DM; prompt template desc; vocabulary $\mathcal{V}$; similarity threshold τ; candidates m; samples per candidate N_w; Strategy A synth budget N_A;
Output: trained task models $\{H_t\}_{t=1}^{K}$.
for $t = 1$ **to** K **do**
  /* Step 1: Strategy A training on task t*/
  $\mathcal{D}_t^{\mathrm{synA}} \leftarrow \emptyset$
  for each class $y \in \mathcal{Y}_t$ **do**
    $\{(\tilde{x}_i, y)\}_{i=1}^{N_A} \leftarrow \mathsf{DM}(\mathrm{concat}(\mathrm{desc}, \mathrm{name}(y)))$
    $\mathcal{D}_t^{\mathrm{synA}} \leftarrow \mathcal{D}_t^{\mathrm{synA}} \cup \{(\tilde{x}_i, y)\}_{i=1}^{N_A}$
  end for
  Train classifier H_t^A on $(\mathcal{D}_t \cup \mathcal{D}_t^{\mathrm{synA}})$ using CE loss
  /* Step 2: Select auxiliary token per class*/
  $\mathcal{D}_t^{\mathrm{aux}} \leftarrow \emptyset$
  for each class $y \in \mathcal{Y}_t$ **do**
    $\mathcal{W}_y \leftarrow \mathrm{SampleTokens}(\mathcal{V}, \mathrm{name}(y), m, \tau)$
    {/* Sample only tokens w satisfying $\mathrm{sim}(w, y) = \langle \bar{e}(w), \bar{e}_y \rangle \leq \tau$.*/}
    $W_y^* \leftarrow \texttt{null}; \quad L_{\min} \leftarrow +\infty$
    for each candidate $w \in \mathcal{W}_y$ **do**
      $\{(\tilde{x}_{w,i})\}_{i=1}^{N_w} \leftarrow \mathsf{DM}(\mathrm{concat}(\mathrm{desc}, w))$
      $\mathcal{L}_{\mathrm{CE}}(w) \leftarrow -\frac{1}{N_w} \sum_{i=1}^{N_w} \log \sigma_t^A(\tilde{x}_{w,i}, y)$
      if $\mathcal{L}_{\mathrm{CE}}(w) < L_{\min}$ **then**
        $L_{\min} \leftarrow \mathcal{L}_{\mathrm{CE}}(w); \quad W_y^* \leftarrow w$
      end if
    end for
    Introduce auxiliary label $y_{\mathrm{aux}}(y)$ (one-to-one)
    $\{(\tilde{x}_i, y_{\mathrm{aux}}(y))\}_{i=1}^{N_A} \leftarrow \mathsf{DM}(\mathrm{concat}(\mathrm{desc}, W_y^*))$
    $\mathcal{D}_t^{\mathrm{aux}} \leftarrow \mathcal{D}_t^{\mathrm{aux}} \cup \{(\tilde{x}_i, y_{\mathrm{aux}}(y))\}_{i=1}^{N_A}$
  end for
  /* Step 3: Train task-t model with CMDA */
  Train H_t on $\mathcal{D}_t \cup \mathcal{D}_t^{\mathrm{synA}} \cup \mathcal{D}_t^{\mathrm{aux}}$ by minimizing
    $\mathbb{E}_{(x,y) \sim \mathcal{D}_t \cup \mathcal{D}_t^{\mathrm{synA}} \cup \mathcal{D}_t^{\mathrm{aux}}} \mathcal{L}_{\mathrm{CE}}(H_t(x), y)$
end for

A.3 Algorithm

The step-by-step implementation of our method is provided in Algorithm 1.

A.4 Additional Implementation Details

This section provides the details of the continual learning classifier and the generation using the diffusion model. The source code is available here:

https://github.com/ritmininglab/CMDA-NeurIPS2026

Classifier Detail: We use ResNet18 for the CUB200 and ImageNet datasets and ResNet32 for CIFAR100. We set Learning rate of 0.05, the number of epochs of 200, and the batch size of 128 for each task learning. While training each task, a gradient reversal layer with binary cross-entropy loss to distinguish between real and synthetic images is also employed, which helps in closing the distribution gap between real and synthetic data. During the classifier update step, the learning rate is set to 0.001 and the number of epoch to 20 for the ImageNet dataset, where as 0.1 learning rate with 3 epochs for the CIFAR100 and CUB datasets. In this step, a knowledge distillation (KD) loss is also employed with a 0.7 coefficient for KD loss for CUB and CIFAR100 datasets and 0.8 for the ImageNet dataset

Table 5: Candidate-budget stability on Split-CUB. Agreement and selected CE increase are measured relative to $m = 20$, with five generations per candidate.

	Token agreement (%)		CE increase	
m	SD1.5	Würstchen v2	SD1.5	Würstchen v2
5	34.0	26.5	0.719	0.779
10	53.0	51.5	0.350	0.335
15	76.0	76.5	0.177	0.155
20	100.0	100.0	0.000	0.000

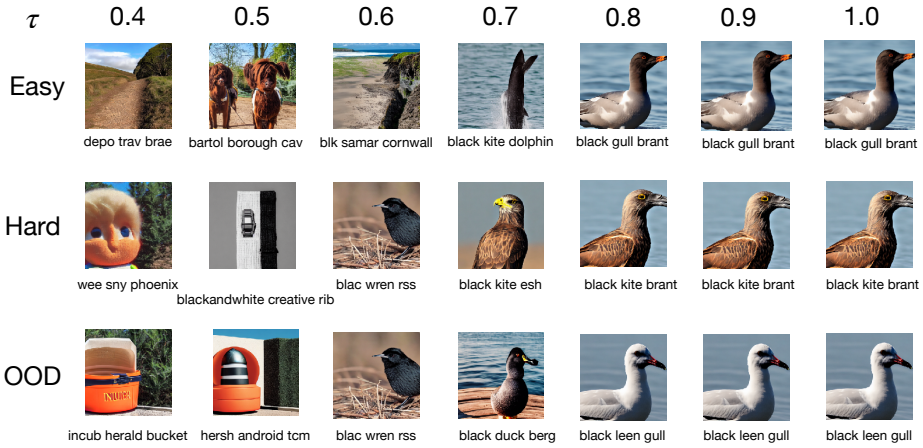


Figure 4: Generated examples selected for the CUB class "Black-footed Albatross." Columns vary the similarity threshold τ; rows show EASY (low CE), HARD (high CE), and OOD selection. The strings below the images are the selected token sequences. The prompt template and generation seeds are held fixed.

Generation Detail: We use Stable Diffusion V1.5 for the Diffusion Model. The number of diffusion steps is 25, and the guidance scale is 7.5. We sample 20 tokens per similarity threshold τ and generate 5 samples per token. We sample an equal number of synthetic data as real data per class: 500 images per class for CIFAR100, 60 images per class for CUB200, and 1300 images per class for ImageNet100. The extra auxiliary synthetic classes also have the same number of samples per class. We vary τ from 0.4 to 1 as the similarity region in extreme values is very sparse, such that tokens with similarity less than 0.4 are very rare.

Token-selection stability. Here we study token-selection stability isolated from downstream CIL performance. On Split-CUB, fixing the candidate list and using one generation per candidate gives 13.9% pairwise exact-token agreement across five generation seeds. With our default five generations per candidate, leave-one-generation-out selection (which checks whether omitting one generated image changes the selected token) retains the token selected using all five generations 72.0% of the time.

We also compare two diffusion backbones, Stable Diffusion v1.5 (SD1.5) [24] and Würstchen v2 [22], with their respective text encoders. The selected tokens compared between the backbones agree for 19.0% of classes, indicating that token identity varies across generator–encoder configurations.

Similarly, Table 5 examines sensitivity to the candidate budget. We retain $m = 20$ as the default and compare with decreasing m. Using $m = 15$ retains the default selection for 76.0%/76.5% of classes under SD1.5/Würstchen v2, while reducing the number of images generated for candidate evaluation by 25%.

Table 6: Split-CUB CIL accuracy (%) before and after CMDA for different generator–encoder configurations.

Diffusion backbone	Text encoder	Before	After
Stable Diffusion v1.5	CLIP ViT-L/14	19.34	32.23
SANA-Sprint 0.6B	Gemma 2 2B-IT	19.04	27.23
Würstchen v2	CLIP ViT-bigG/14	23.45	34.44

Multi-token class names and beam search under a threshold. Many class names consist of multiple text tokens (*e.g.,* `black_footed_albatross`). To obtain a single class embedding, we represent the class name by a length-K_y token sequence $(u_{y,1}, \ldots, u_{y,K_y})$ and use the normalized mean embedding

$$e_y := \frac{1}{K_y} \sum_{k=1}^{K_y} \psi(u_{y,k}), \qquad \bar{e}_y := \frac{e_y}{\|e_y\|_2}. \tag{30}$$

In practice, we construct candidate *phrases* of length K_y by beam search under the same similarity constraint. Specifically, beam search restricts each partial hypothesis to tokens satisfying $\mathrm{sim}(w, y) \leq \tau$ and selects the top candidates by maximizing the average similarity to the class embedding:

$$\begin{aligned} \mathrm{score}(w_{1:K_y}) \;&:=\; \frac{1}{K_y} \sum_{k=1}^{K_y} \langle \bar{e}(w_k),\, \bar{e}_y \rangle, \\ \text{s.t. } \mathrm{sim}(w_k, y) &\leq \tau \;\; \forall k \end{aligned} \tag{31}$$

We then treat each selected phrase as a tokenized prompt modifier and evaluate it via $\mathcal{L}_{\mathrm{CE}}$ as above.

Qualitative examples. Figure 4 illustrates how the selected tokens and generated content vary with the similarity threshold and selection rule. For example, EASY selects "blk samar cornwall" at $\tau = 0.6$, producing a coastal scene, and "black gull brant" at $\tau = 0.8$, producing a waterbird-like image. These generations may retain contextual cues, such as water or shoreline, without reproducing the target class. Such overlap illustrates the motivation for assigning separate auxiliary labels. However, these visual examples do not establish which cues the classifier actually uses or whether training reduces its reliance on them. The OOD variant selects tokens using a low-summed-logit criterion.

Figure 5 compares prediction corrections (CMDA helped) and failures (CMDA did not help) using Grad-CAM visualization. For the class 'Cardinal', the selected token is `apple` (candidate examples: `morning`, `court`, `apple`); for class 'Indigo Bunting', it is `bluec canary` (candidate examples: `bluec obo`, `purple tern`, `bluec canary`). The generated images share red or blue appearance with the respective birds, suggesting possible feature overlap. In the corrected examples, CMDA's maps show greater emphasis over the bird's body. However, the same tokens also accompany prediction failures on other images. These examples illustrate attribution differences, rather than directly verifying shortcut removal.

A.5 Additional Experiments

A.5.1 Robustness across generator–encoder pairs.

We evaluate CMDA on Split-CUB using three diffusion backbones and their associated text encoders: Stable Diffusion v1.5 [24], SANA-Sprint 0.6B [4], and Würstchen v2 [22]. Table 6 compares the corresponding baseline before and after applying CMDA.

Although absolute accuracy varies across configurations, CMDA improves performance within every tested pair. This provides evidence of functional robustness beyond Stable Diffusion v1.5. Since the generator and text encoder change together, the comparison does not isolate the effect of the vocabulary alone.

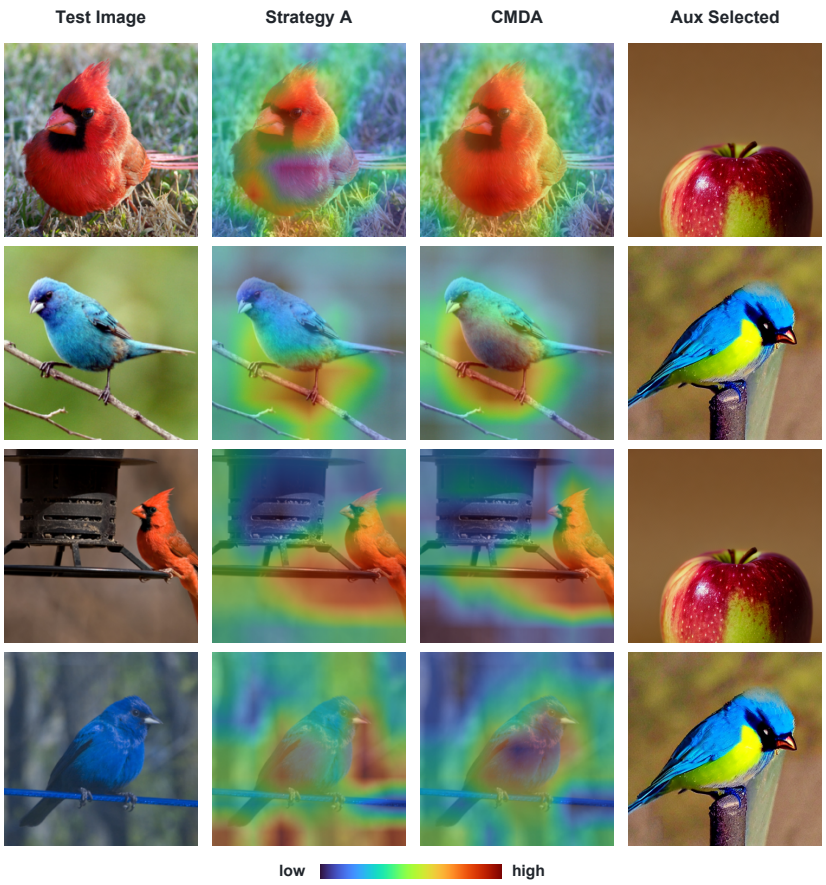


Figure 5: Qualitative Split-CUB examples. Rows 1 and 3 show class 'Cardinal' with the selected token `apple`; rows 2 and 4 show class 'Indigo Bunting' with `bluec canary`. In the top two rows, CMDA changes incorrect predictions to correct ones; in the bottom two, it changes correct predictions to incorrect ones. Warm colors indicate higher attribution.

Table 7: Effect of token CE loss on Split-CUB. Across all similarity thresholds τ, lower-loss tokens consistently yield better final accuracy than higher-loss tokens.

τ	Token Type	CE Loss	Acc.
0.5	Low-loss	2.64	32.15
0.5	High-loss	9.87	30.83
0.6	Low-loss	2.24	30.75
0.6	High-loss	10.41	27.65
0.7	Low-loss	1.76	32.23
0.7	High-loss	9.97	29.55
0.8	Low-loss	1.56	32.02
0.8	High-loss	9.12	25.78
0.9	Low-loss	1.43	32.44
0.9	High-loss	8.72	30.19

A.5.2 Empirical support for the token-selection assumption.

Directly verifying Assumptions A.1 and A.2 is difficult because we do not have oracle annotations indicating whether a candidate token is truly shortcut-breaking. Thus, our theory should be interpreted as conditional on the token-selection signal being informative. The key observable implication is that tokens with lower cross-entropy (CE) under the Strategy-A classifier should be more useful

Table 8: Multi-seed comparison between RANDOM and FAR EASY.

Method	Split-CUB	Split-CF100
RANDOM	30.15 ± 0.97	56.15 ± 0.13
FAR EASY	32.19 ± 0.19	57.15 ± 0.25

than tokens with higher CE. We test this implication on Split-CUB by comparing low-loss and high-loss selected tokens across different similarity thresholds τ. As shown in Table 7, lower-CE tokens consistently lead to higher final accuracy than higher-CE tokens across all thresholds. These results support CE as a useful ranking signal for auxiliary-token selection, but do not directly verify assumptions A.1 or A.2.

Table 8 reports multi-seed results for RANDOM and FAR EASY. FAR EASY achieves higher mean CIL accuracy on both datasets, supporting the additional benefit of classifier-guided token selection. The standard deviations are obtained from three runs with p-values of 0.045 and 0.005 respectively.

A.5.3 Feature concentration.

We examine whether CMDA distributes classifier relevance across more penultimate feature dimensions. Following the score used in FLAP [2], we construct importance scores (activation-variance multiplied by squared-weight importance score) for penultimate feature dimensions using fixed real-image reference sets and centered real-class classifier weights. The analysis covers all ten tasks on Split-CF100.

We measure the concentration of the resulting nonnegative importance vector $s \in \mathbb{R}^d$ using Hoyer sparsity [11]:

$$H(s) = \frac{\sqrt{d} - \|s\|_1/\|s\|_2}{\sqrt{d} - 1}, \qquad s \neq 0.$$

Lower values indicate that importance is distributed across more feature dimensions, while higher values indicate more concentrated feature use.

Hoyer sparsity decreases from 0.342 under Strategy A to 0.277 under CMDA, a difference of -0.065 with a 95% confidence interval of $[-0.120, -0.013]$. This is consistent with less concentrated classifier reliance. It provides an indirect diagnostic of the proposed mechanism, rather than identifying which features are shortcuts or establishing a causal explanation for the accuracy gains.

A.5.4 Controlled planted-shortcut experiment.

We construct a controlled setting in which the shortcut is known by design. We split CUB-200 into ten tasks of twenty classes using the original class order, and group the tasks into five fixed pairs: $(0, 1), (2, 3), \ldots, (8, 9)$. Every real and synthetic image receives a fixed 28×28 corner patch. Its border color encodes the task pair, and its interior color encodes the within-task class index. The same patch is used for corresponding class indices in both tasks of a pair. Thus, the patch identifies the within-task class but cannot distinguish the two tasks. Under equal task priors, a classifier relying only on this shortcut has a pairwise task-ID accuracy ceiling of 50%.

Matched training conditions. We compare three conditions. Strategy A uses real images and sixty same-class Stable Diffusion v1.5 images per class. SAME adds sixty FAR EASY images per class, assigning them the original class label. AUX uses exactly the same images and sample order as SAME, but assigns the FAR EASY images separate auxiliary labels. Consequently, the same patch now corresponds to both a real and an auxiliary label, making the patch alone insufficient to solve the training objective. Class exposure is equalized by deterministic oversampling. Patches are applied after all augmentations so they cannot be cropped out. We use a fixed stratified 10% validation split and retain the original test split.

Pairwise task-ID evaluation. Using only Strategy A predictions on real training data, we identify the confused task pairs. This recovers the five designed pairs and remains unchanged in all twenty leave-one-class-out checks. We freeze these pairs and evaluate whether each model selects the correct task within its pair, with the patch present as during training.

Table 9: Pairwise task-ID accuracy (%) in the controlled planted-shortcut setting. SAME and AUX use identical additional images and differ in their labels.

Condition	Validation	Test
Strategy A	52.3	49.4
SAME	49.3	50.6
AUX	68.3	67.9

Table 10: Intermediate task-wise results on Split-CF100. TIL changes only marginally, while TIP improves consistently from Task 2 onward. The CIL gains follow the TIP trend, suggesting that CMDA mainly improves task-ID prediction.

Metric	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
TIL Before	93.60	88.85	90.20	89.83	90.60	90.18	90.57	90.25	90.23	89.78
TIL After	92.70	88.55	90.40	90.05	90.84	90.52	90.71	90.48	90.50	90.09
TIP Before	–	83.75	77.53	71.65	67.42	63.68	62.43	59.03	56.92	55.87
TIP After	–	87.60	80.47	74.28	71.30	67.93	66.24	63.45	61.48	59.25
CIL Before	93.60	78.10	74.10	68.60	65.24	61.67	60.77	57.39	55.46	54.35
CIL After	92.70	80.65	76.40	70.90	68.68	65.58	64.30	61.53	59.64	57.39

As shown in Table 9, Strategy A and SAME remain near 50%, whereas AUX reaches 67.9% test accuracy. Because the patch cannot distinguish tasks within a pair, the higher accuracy requires information beyond that shortcut. This supports the role of auxiliary labels in reducing reliance on the specific planted cue. The experiment isolates auxiliary labeling rather than minimum-CE selection, and does not establish robustness to arbitrary natural shortcuts.

A.5.5 Intermediate task-wise analysis.

We further report TIL, TIP, and CIL accuracy at each intermediate continual learning step on Split-CF100. This allows us to examine whether CMDA improves performance by strengthening the per-task classifiers or by improving task-ID prediction. As shown in Table 10, TIL accuracy changes only marginally across tasks, while TIP improves consistently from Task 2 onward. The CIL gains follow a similar trend to TIP, consistent with improved task separation. These results support the proposed mechanism, but do not establish that reduced shortcut reliance causes the accuracy gains on natural benchmarks.

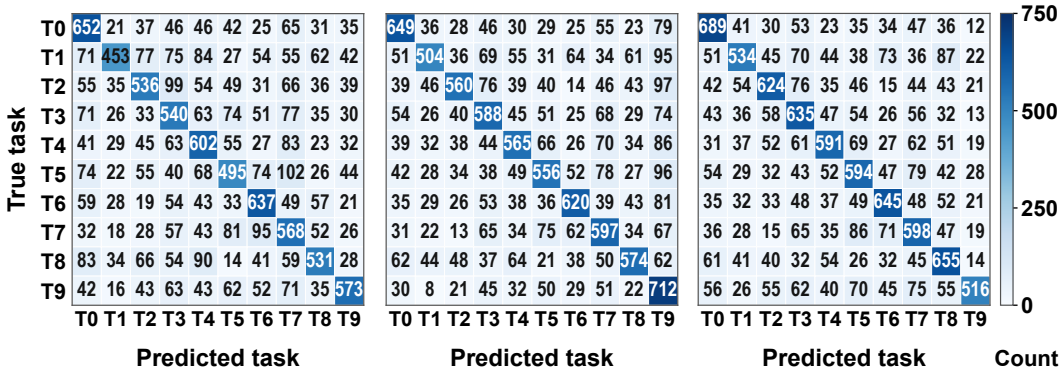
A.5.6 Task routing and optional calibration.

Maximum-logit routing can be biased by differences in logit scale across task-specific classifiers. CMDA improves task routing, but residual bias can remain. We therefore evaluate an optional exemplar-free calibration step that re-updates the task-specific classifiers using zero-shot synthetic images from previous tasks.

Table 11 separates the improvement from CMDA from the additional calibration gain. Before calibration, CMDA improves TIP by 3.38 percentage points, with task-wise TIP improving for seven of ten tasks. Calibration provides a further 1.56 points and narrows the range of task-wise TIP accuracies. Figure 6 shows the task-ID confusion matrices for Strategy A, CMDA, and CMDA with calibration.

A.5.7 Evaluation under domain-mismatched diffusion generation.

We further evaluate CMDA on Split-BloodMNIST, an out-of-domain medical benchmark where shortcut learning can be particularly severe. This setting is challenging for zero-shot diffusion augmentation because the frozen generator may not faithfully cover specialized medical image distributions. Importantly, CMDA does not require the diffusion model to reproduce the full target data distribution. Instead, it selects classifier-guided auxiliary classes whose generations only need to partially overlap with shortcut-relevant regions of the target class. Such samples can still act as shortcut-breaking examples and reduce shortcut reliance during training.



(a) Strategy A (b) CMDA (c) CMDA + Calibration

Figure 6: Task-ID confusion matrices on Split-CF100. Rows denote true tasks and columns denote predicted tasks, with 1,000 test images per true task.

Table 11: Split-CF100 task-id prediction results. All values are percentages; minimum and maximum TIP are taken across the ten tasks.

Method	CIL	TIP	Min TIP	Max TIP
Strategy A, raw	54.36	55.87	45.3	65.2
CMDA, raw	57.39	59.25	50.4	71.2
CMDA + calibration	58.99	60.81	51.6	68.9

Table 12: Results on Split-BloodMNIST. CMDA improves both a from-scratch ResNet setting and a stronger pretrained ViT setting with LoRA-DRS.

Backbone	Setting	Baseline	+CMDA	Δ
ResNet	From scratch	43.38	45.72	$+2.34$
ViT	LoRA-DRS	74.12	79.14	$+5.02$

Table 12 shows that CMDA improves performance on Split-BloodMNIST [5] in both settings. With a from-scratch ResNet, CMDA improves accuracy from 43.38 to 45.72, yielding a gain of $+2.34$ points. With a pretrained ViT and the stronger LoRA-DRS baseline, CMDA improves accuracy from 74.12 to 79.14, yielding a larger gain of $+5.02$ points. The diffusion generator and its vocabulary are fixed in both settings, while the continual learner changes from a from-scratch ResNet to a pretrained ViT with LoRA-DRS. The larger gain with the pretrained learner may reflect a more informative CE signal and a better ability to exploit the auxiliary samples. This comparison therefore does not isolate the effect of generator coverage. CMDA remains dependent on the frozen generator's vocabulary and visual priors, which may limit the usefulness of its auxiliary samples in specialized domains.

A.5.8 Compatibility with other continual learning methods.

To verify that CMDA is not limited to the original from-scratch backbone setting, we additionally evaluate it with a pretrained ViT-B/16 backbone on Split-CUB and compare against recent prompt- and adapter-based EFCIL baselines. As shown in Table 13, CMDA improves the strong LoRA-DRS [20] baseline from 88.29 to 88.53 average accuracy. Moreover, LoRA-DRS+CMDA remains competitive with CODA-Prompt [33], FeCAM [9], and ADC [10]. These results indicate that CMDA is complementary to stronger pretrained EFCIL methods: rather than replacing prompt- or adapter-based adaptation, CMDA can further improve them by providing shortcut-breaking auxiliary classes during training.

We further evaluate compatibility with replay- and prompt-based learning on Split-CUB. In the replay setting, we use a class-balanced buffer of 200 real images alongside current-task real and same-class synthetic data. Adding CMDA improves final CIL accuracy from 60.64% to 62.26%.

Table 13: Comparison with pretrained EFCIL baselines on Split-CUB using a ViT-B/16 backbone. CMDA further improves the strong LoRA-DRS baseline.

Method	Avg. Accuracy
CODA-Prompt	84.764
FeCAM	86.861
ADC (LoRA)	88.06 ± 0.015
LoRA-DRS	88.29 ± 0.04
LoRA-DRS + CMDA	$\mathbf{88.53 \pm 0.13}$

Table 14: Results on few-shot and long-tailed Split-CF100. CMDA remains effective under data scarcity and class imbalance.

Setting	Before Aug.	After Aug.	Δ
Few-shot	36.78	48.43	$+11.65$
Long-tailed	65.31	74.78	$+9.47$

In a separate prompt-based setting, we use a pretrained ViT-B/16 with a CODA-Prompt-based learner [33], a prompt pool of 100, and ten prompts allocated per task. We report mean CIL accuracy over the seen tasks, averaged across ten incremental stages and three seeds. An adapted CMDA configuration increases this continual-average accuracy from 76.52% to 77.53%. This configuration disables domain-adversarial alignment, weights the CE contributions of real, same-class synthetic, and auxiliary samples as 0.50/0.25/0.25, and applies two real-only calibration epochs after each task.

These results provide preliminary evidence that CMDA auxiliary augmentation can be integrated with replay- and prompt-based learning. Broader evaluations across these methods remain future work.

A.5.9 Comparison with general augmentation methods.

We additionally compare CMDA with Cap2Aug [25], a caption-guided same-class augmentation method. As shown in Table 15, CMDA outperforms Cap2Aug by $+1.86$ points on Split-CF100 and $+6.39$ points on Split-CUB. We attribute this improvement to the difference in how synthetic data is used. Cap2Aug augments each original class with caption-guided synthetic images from the same class, making it most comparable to the same-class augmentation variants in Table 2. In contrast, CMDA uses selected generations as auxiliary classes, which directly targets shortcut collapse by encouraging stronger task separation. This result is consistent with our ablation study, where same-class generation variants such as EASY SAME are consistently weaker than auxiliary-class augmentation.

Table 15: Comparison with Cap2Aug. CMDA outperforms caption-guided same-class augmentation.

Dataset	Cap2Aug	CMDA	Δ
Split-CUB	25.83	32.23	$+6.39$
Split-CF100	55.53	57.39	$+1.86$

We also compare with DiffCoRe-Mix [12] and FracMix [13], adapted to our continual learning setting. Table 16 reports the Split-CUB results under the from-scratch protocol. CMDA achieves the highest CIL accuracy, improving over DiffCoRe-Mix and FracMix by 16.37 and 1.95 percentage points, respectively, while FracMix achieves higher TIL accuracy than CMDA.

A.5.10 Generality under data scarcity and class imbalance.

We further evaluate the generality of CMDA under two challenging class-incremental learning settings on Split-CF100: few-shot CIL and long-tailed CIL. In the few-shot setting, each class contains only 5 real training samples. In the long-tailed setting, class frequencies follow an exponential decay and are randomly distributed across tasks, with the largest class containing 500 samples and the smallest class containing 5 samples. We generate 50 synthetic samples per class for the few-shot setting and 100 synthetic samples per class for the long-tailed setting.

Table 14 shows that CMDA remains beneficial when continual learning becomes harder due to limited data or class imbalance. In the few-shot setting, CMDA improves accuracy from 36.78 to 48.43, yielding a gain of $+11.65$ points. In the long-tailed setting, CMDA improves accuracy from 65.31 to 74.78, yielding a gain of $+9.47$ points. The larger gain in the few-shot regime suggests that

Table 16: CIL and TIL accuracy (%) on Split-CUB under the from-scratch setting.

Method	CIL	TIL
DiffCoRe-Mix	15.86	48.60
FracMix	30.28	**59.57**
CMDA	**32.23**	58.14

Table 17: Accuracy and per-task cost on Split-CUB. Aux denotes auxiliary classes per real class. Times are in hours per task; relative cost is normalized to Aux = 1.

Aux	CIL (%)	TIP (%)	Generation + selection	CL training	Total	Relative cost
1	32.23	44.19	1.2	0.3	1.5	1.0×
3	38.80	50.64	3.0	0.4	3.4	2.3×
5	40.71	53.44	4.8	0.6	5.4	3.6×

shortcut-breaking auxiliary augmentation is especially useful when the classifier is more likely to overfit to limited task-specific cues. The improvement in the long-tailed setting further indicates that CMDA can provide useful additional supervision even when the training distribution is highly imbalanced.

A.5.11 Effect of using multiple auxiliary classes.

Table 17 shows that increasing the number of auxiliary classes improves both CIL and TIP, while increasing generation and training costs. The accuracy gains diminish at larger budgets: increasing Aux from 1 to 3 adds 6.57 CIL points, whereas increasing it from 3 to 5 adds a further 1.91 points. Meanwhile, total cost rises from 1.5 to 3.4 and 5.4 hours per task, respectively. We therefore use Aux = 1 as our default low-budget configuration. Larger auxiliary budgets provide higher accuracy when additional compute is available.

Table 18 compares selection strategies using the same auxiliary-label and synthetic-sample budgets within each column. All strategies improve as the auxiliary budget increases. FAR EASY achieves the highest observed accuracy at each budget, but its advantage over RANDOM narrows from 1.70 percentage points at Aux = 1 to 0.56 and 0.52 points at Aux = 3 and 5. This suggests that broader random auxiliary augmentation recovers much of the benefit when more generation is allowed, while classifier-guided selection provides its largest additional gain under the default limited budget. We focus this comparison on strategies sharing the same auxiliary-label construction, since Aux is not a shared hyperparameter across external EFCIL methods.

A.5.12 Effect of gradient reversal.

We use gradient reversal layer (GRL) mainly for consistency with the DiffClass-style training setup, where it encourages alignment between real and synthetic features and reduces the real–synthetic feature gap. However, CMDA is not highly sensitive to this component. As shown in Table 19, removing GRL causes only a small drop on Split-CF100, from 58.95 to 58.67. We attribute this robustness to the CE-based token filtering step: CMDA selects auxiliary tokens whose generations already overlap well with the current classifier, so GRL provides only a modest additional alignment benefit.

Table 19: Ablation of GRL on Split-CF100. CMDA is not highly sensitive to GRL.

Dataset	w/o GRL	w/ GRL
Split-CF100	58.67	58.95

A.5.13 Computational efficiency.

From Table 17, on Split-CUB, classifier-only CL training takes approximately 0.3 hours per task. With one auxiliary class per real class, CMDA has an estimated task-update cost of approximately 1.5 hours: 1.2 hours for image generation, including candidate images used in token selection, and 0.3 hours for CL training. This corresponds to approximately five times the classifier-only cost. Training was performed on NVIDIA RTX A6000, with batch size 128.

Table 18: Token-selection strategies at fixed auxiliary budgets on Split-CUB. Values are CIL accuracy (%). Aux denotes auxiliary classes per real class.

Strategy	Aux = 1	Aux = 3	Aux = 5
RANDOM	30.53	38.24	40.19
FAR HARD	30.83	37.16	39.48
NEAR EASY	20.10	33.77	34.64
FAR EASY	**32.23**	**38.80**	**40.71**

Table 20: Per-task generation cost on Split-CUB. CMDA avoids diffusion-model training and therefore reduces total generation time.

Method	Sample Gen. (s)	DM Train. (s)	Total (s)
DiffClass	4680	2617	7297
CMDA	4400	0	4400

CMDA is designed to add only a modest overhead over vanilla diffusion augmentation, while remaining substantially cheaper than methods that adapt the diffusion model or rely on gradient-guided generation. Appendix A.6 provides the full cost model. Here, we report concrete per-task computational costs on Split-CUB and compare CMDA with DiffClass.

As shown in Table 20, CMDA requires less total generation time because it avoids diffusion-model training. The classifier parameter counts are nearly identical across methods; the main parameter-efficiency advantage comes from the fact that CMDA does not introduce trainable diffusion-model parameters (Table 21). Moreover, generation is needed only during task updates. At inference or deployment time, the auxiliary classes are discarded, so CMDA introduces no additional inference-time cost.

Table 21: Trainable parameter comparison on Split-CUB. CMDA avoids diffusion-model adaptation and uses fewer trainable parameters.

Method	DM LoRA	Classifier	Total
DiffClass	1,594,368	11,264,009	12,858,377
CMDA	0	11,253,609	11,253,609

A.6 Computational Cost Analysis

We distinguish *training-time* cost (optimizing model parameters) from *generation-time* cost (synthesizing images at inference or data-creation time).

(1) Training-time cost: diffusion adaptation vs. Strategy A vs. Strategy A+M. Let $C_{\text{DM-train}}$ denote the cost to train a diffusion model from scratch (e.g., Stable Diffusion), and $C_{\text{DM-ft}}$ the cost to adapt a pretrained diffusion model (e.g., DreamBooth, LoRA, or prompt-tuning). These costs are dominated by backpropagation through the diffusion denoiser across multiple timesteps and thus scale with the number of update steps and timesteps:

$$C_{\text{DM-train}} = \Omega(U_{\text{DM}} \cdot n \cdot c_{\text{step}}),\ C_{\text{DM-ft}} = \Omega(U_{\text{ft}} \cdot n \cdot c_{\text{step}}),$$

where U_{DM} (resp. U_{ft}) is the number of optimization steps, n is the number of diffusion steps per sample, and c_{step} is the per-step denoising cost (forward+backward).

In contrast, Strategy A trains only the continual learner on task t using real data and *vanilla* diffusion samples. Let $C_{\text{CL-train}}$ denote the cost per optimizer step for the continual learner and U_t the number of SGD steps for task t. Then

$$C_{\text{A}} = \Theta(U_t \cdot C_{\text{CL-train}}),$$

where the diffusion model is used only for forward sampling (no gradient updates to the diffusion model).

CMDA adds a token selection stage but still does *not* update the diffusion model. For serial execution, the full task-update cost comprises

$$C_{\text{CMDA}}(t) = \sum_{y \in \mathcal{Y}_t} \big(N_{t,y} + N_{t,y}^{\text{aux}}\big)v + C_{\text{select}}(t) + C_{\text{A}}(t) + C_{\text{aux-train}}(t).$$

Here, $C_{\mathrm{A}}(t)$ is the cost of training the Strategy A classifier used for selection, and $C_{\text{aux-train}}(t)$ is the subsequent CMDA classifier-training cost, including any recalibration actually applied. If classifiers or synthetic datasets are reused, their preparation costs should be reported separately or amortized explicitly.

(2) Generation-time cost: vanilla sampling vs. CMDA vs. gradient-guided generation. Let v denote the wall-clock cost to generate one image via standard diffusion sampling (e.g., DDIM) using n steps. Strategy A uses vanilla diffusion generation, so generating $N_{t,y}$ samples for class y costs

$$C^{\mathrm{A}}_{\mathrm{gen}}(y) = N_{t,y}\, v.$$

Cost of CMDA token/phrase selection. For class y, CMDA evaluates m candidate tokens/phrases, each by generating N_w images and scoring them with the Strategy A classifier. Let $c_{\text{clf-fwd}}$ be the cost of one forward pass through H_t^A. Then the selection cost for class y is

$$C_{\mathrm{select}}(y) = \underbrace{mN_w\, v}_{\text{diffusion sampling}} + \underbrace{mN\, c_{\text{clf-fwd}}}_{\text{classifier scoring}}.$$

For a task with $C_t = |\mathcal{Y}_t|$ incoming classes, selection requires $C_t m N_w$ candidate images and the same number of image-level classifier evaluations:

$$C_{\mathrm{select}}(t) = C_t m N_w \big(v + c_{\text{clf-fwd}}\big).$$

Thus, the candidate-generation count scales linearly with C_t for fixed m and N_w. Candidate generation can be parallelized, and classifier scoring can be batched.

After selecting w_y^*, generating the auxiliary dataset of size $N^{\mathrm{aux}}_{t,y}$ costs an additional $N^{\mathrm{aux}}_{t,y}\, v$. Therefore, the total generation-time overhead of CMDA for class y is

$$C^{\mathrm{M}}_{\mathrm{gen}}(y) = mN_w\, v + mN_w\, c_{\text{clf-fwd}} \;+\; N^{\mathrm{aux}}_{t,y}\, v.$$

Gradient-guided diffusion generation. An alternative to token search is to guide generation by backpropagating gradients from a classifier to (i) the latent trajectory or (ii) the text/prompt embedding, or even (iii) the diffusion weights. Let $v_{\mathrm{g,lat}}$ be the cost to generate one sample when gradients are computed w.r.t. the latent (requires backprop through the denoiser across timesteps), and $v_{\mathrm{g,tok}}$ when gradients are computed w.r.t. the token/prompt embedding (often more expensive due to additional conditioning and optimization steps). Typically,

$$v_{\mathrm{g,tok}} \;\gtrsim\; v_{\mathrm{g,lat}} \;\gg\; v,$$

since gradient guidance requires backward passes (and often multiple inner-loop update steps) during sampling. The per-image cost depends on the guided sampler, including its backward passes and inner-loop updates.

Thus, generating $N_{t,y}$ guided samples for class y costs

$$C^{\mathrm{grad}}_{\mathrm{gen}}(y) = N_{t,y}\, v_{\mathrm{g},\cdot} \quad \text{with} \quad v_{\mathrm{g},\cdot} \in \{v_{\mathrm{g,lat}},\, v_{\mathrm{g,tok}}\}.$$

Summary of generation-time comparisons. Strategy A is cheapest per image (v). CMDA adds a one-time selection overhead $mN(v + c_{\text{clf-fwd}})$ per class, but still relies on *vanilla* sampling thereafter. In contrast, gradient-guided generation pays a higher per-image cost $v_{\mathrm{g},\cdot}$ for every generated sample. Consequently, for moderate mN (small token-search budget) and non-trivial $N_{t,y}$, CMDA is typically more efficient than gradient-guided generation: $C^{\mathrm{M}}_{\mathrm{gen}}(y) \;=\; mN(v + c_{\text{clf-fwd}}) + N^{\mathrm{aux}}_{t,y} v \;\ll\; N_{t,y}\, v_{\mathrm{g},\cdot} \;=\; C^{\mathrm{grad}}_{\mathrm{gen}}(y)$, while still being more expensive than pure vanilla sampling $C^{\mathrm{A}}_{\mathrm{gen}}(y) = N_{t,y} v$.

At equal retained-image budgets, CMDA has lower generation and selection cost than a gradient-guided implementation when

$$mN_w\big(v + c_{\text{clf-fwd}}\big) < N^{\mathrm{aux}}_{t,y}\big(v_{\mathrm{g},\cdot} - v\big).$$

The comparison therefore depends on the candidate budget, the number of retained images, and the guided sampler's per-image cost. Avoiding diffusion-model optimization can reduce cost, but total runtime still depends on the generation budget and the method being compared.

B Impact Statement and Limitations

This work studies how pretrained diffusion models can be used to support EFCIL. At the same time, diffusion-based augmentation may inherit or amplify biases present in pretrained generative models and can introduce distribution shifts if synthetic data is misaligned with the target domain. We therefore emphasize that CMDA does not generate new sensitive attributes by design and should be deployed with standard safeguards, including bias evaluation, dataset auditing, and monitoring for performance regressions under distribution shift. CMDA also incurs additional offline generation and task-update costs, and its effectiveness depends on the frozen generator's visual priors and domain coverage, the similarity threshold, and generated-image quality.

CMDA also depends on the text side of the diffusion model because candidate auxiliary classes are sampled from the model vocabulary and filtered in its text-embedding space; thus, the available candidates and the effectiveness of "near/far" sampling depend on that vocabulary geometry. Our main results uses Stable Diffusion v1.5 only but we observe improvements on Split-CUB with Stable Diffusion v1.5 (CLIP ViT-L/14), SANA-Sprint 0.6B (Gemma 2 2B-IT), and Würstchen v2 (CLIP ViT-bigG/14). These results cover three generator–encoder pairs and do not isolate vocabulary effects or establish robustness to arbitrary domain shifts. At the same time, the method is not CLIP-specific in principle: it only requires a text-conditioned generator with an accessible text embedding space plus image generation from candidate prompts. We note that evaluating other diffusion backbones/tokenization schemes and extending the search from tokens to phrases is an important future direction. Our theoretical guarantees are conditional on idealized assumptions. Natural-data accuracy and feature-concentration results provide indirect evidence of shortcut mitigation.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide impact and limitations in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We provide the detailed assumptions and proof in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the complete source code in the supplementary and the algorithm and additional experiment details in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the source code link in the Appendix, and the datasets used are publicly available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the details in both the main paper and the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report the 1-sigma error for our result in the main table.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed and comply with the guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the negative impact and limitations in the appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Although we use a diffusion model, the standard safeguard for this already exists for publicly available models. We also discuss this as a limitation in the appendix.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use standard datasets and models that are publicly available.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not introduce any new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: We do not perform any crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We do not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We do not use LLMs for the core method.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.