PromptCCD: Learning Gaussian Mixture Prompt Pool for Continual Category Discovery
–*Supplementary Material*–

Fernando Julio Cendra[1], Bingchen Zhao[2], and Kai Han[1†]

[1]The University of Hong Kong [2]The University of Edinburgh

We provide this supplementary material to further support our main paper. We begin with a pseudo-code implementation of our method in Sec. S1. Next, in Sec. S2, we present the breakdown CCD results, benchmarking the performance of each compared model across different stages for cases when number of categories C are known and unknown. Additionally, we delve into inductive evaluation scenarios (Sec. S3), evaluation on standard GCD metric (Sec. S4), additional comparison with other CCD settings (Sec. S6), analysis on different class splits scenarios (Sec. S7), and qualitative results (Sec. S9) in separate sections. Our implementation details cover aspects such as our implementation of Grow & Merge to enhance it with ViT in Sec. S5, fine-tuning method and parameter analysis in Sec. S8. Finally, we discuss both the impacts and limitations of our model for future studies in Sec. S10.

Table of Contents

† Corresponding author: Kai Han (kaihanx@hku.hk)

S1 Pseudo Code for PromptCCD *w*/GMP

Algorithm A PromptCCD w/GMP's Pseudo Code.

Require: $\mathcal{H}_\theta : \{\phi, f_\theta\}$ where $f_\theta : \{f_e, f_b\}$.
Require: GMP prompt module where it contains GMM_t.
Require: Dataloader $\mathcal{B}$ for dataset D_t at stage t.

Set $\alpha \leftarrow$ integer value for the incremental update epoch.
Set $\beta \leftarrow$ integer value for the warmup epoch.
procedure PROMPTCCD($\mathcal{H}_\theta$, GMP, $\mathcal{B}$) at stage t.
  /* *********************** *Start training* *********************** */
  for $e \in$ *Epochs* **do**

    /* *fit* GMM_t *every n increment of epoch.* */
    if $0 \equiv e \pmod{\alpha}$ **then**
      $\mathcal{Z}_t \leftarrow \{f_\theta(x) | x \in D_t\}$ // *extract features* **[stop gradient]**
      if $t > 0$ **then**
        $\mathcal{Z}^s_{t-1} \leftarrow$ GENERATE-RANDOM-SAMPLES(GMM_{t-1}).
        $\mathcal{Z}_t \leftarrow \mathcal{Z}_t \cup \mathcal{Z}^s_{t-1}$ // *combine with generated samples from* GMM_{t-1}.
      OPTIMIZE(GMP) by Fitting GMM_t with $\mathcal{Z}_t$

    for $B : \{x_i, x'_i\} \in \mathcal{B}$ **do** // *assume a batch B only contains a set* $\{x_i, x_i^{'}\}$.

      /* *the next lines covered in this box describe how to acquire* $\mu_{top\text{-}k}$. */
      if $e > \beta$ **then** // *when the model reaches the warm-up epoch.*
        $\hat{z}_i \leftarrow f_\theta(x_i)$ // *extract features* **[stop gradient]**.
        $\mu_{\text{top-k}} \leftarrow \text{GMP}(\hat{z}_i | \text{GMM}_t)$ // *see Fig. 3 in main paper for details.*
      else
        $\mu_{\text{top-k}} \leftarrow$ *None*

      /* *the next lines covered in this box describe how* x_i *and* $\mu_{top\text{-}k}$ *are projected into the model [note: same operation for* $x_i^{'}$*].* */
      $x_q \leftarrow \text{PATCHIFY}(x_i)$ // *patchify image* x_i *into L patches.*
      $x_e \leftarrow f_e(x_q)$ // *project to pretrained patch embedding layer.*
      $x_{total} \leftarrow [\mu_{\text{top-k}}; x_e]$ // *concatenate* x_e *with the* $\mu_{top\text{-}k}$ *prompts.*
      $z_i \leftarrow \phi(f_b(x_{total}))$ // *project to self-attention blocks and projection head.*

      /* *to summarize above operations, from* $\mathcal{H}_\theta : \{\phi, f_\theta\}$*, we got:* */
      $z_i \leftarrow \phi(f_\theta(x_i))$ & $z'_i \leftarrow \phi(f_\theta(x'_i))$

    /* *optimize* $\mathcal{H}_\theta$*, (see Sec. 3.1 in main paper) and do* **[gradient update]**. */
    OPTIMIZE($\mathcal{H}_\theta$)
  /* *********************** *End training* *********************** */

S2 Breakdown CCD Benchmark Results

We provide the breakdown results which include the *continual ACC* (*cACC*) (*'All'*, *'Old'*, *'New'*) for each stage following the data splits in [19].

1. Comparison with known class numbers:
    - Table A, comparison on generic datasets with DINO.
    - Table B, comparison on generic datasets with DINOv2.
    - Table C, comparison on fine-grained datasets with DINO.
    - Table D, comparison on fine-grained datasets with DINOv2.
    - Table {E,F,G,H,I}, multiple runs (*5 seeds*) results on variants of PromptCCD with different prompt pool designs on generic and CUB datasets with DINO.
2. Comparison with unknown class numbers, using DINO:
    - Table J, comparison using our GPC-based-estimator [20].
    - Table K, comparison using the k-means-based estimator in [13].

Table A: Breakdown results of various methods for CCD leveraging pretrained DINO model on generic datasets with the *known C* in each unlabelled set.

	Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
CIFAR100	ORCA [1]	62.59	71.55	56.31	63.05	66.38	62.42	57.09	61.90	56.25	60.91	66.61	58.33
	GCD [13]	67.65	83.59	56.49	52.89	68.38	49.93	53.99	64.86	52.08	58.18	72.27	52.83
	SimGCD [17]	35.04	50.65	24.11	22.41	39.05	19.24	19.23	26.57	17.95	25.56	38.76	20.43
	GCD *w*/replay	55.68	80.12	38.57	45.16	67.62	40.87	48.96	71.71	44.98	49.93	73.15	41.47
	SimGCD *w*/replay	48.84	74.16	31.11	35.28	61.43	30.29	36.28	64.57	31.33	40.13	66.72	30.91
	Grow & Merge [19]	64.77	70.49	60.77	58.32	62.95	57.44	49.21	57.62	47.73	57.43	63.68	55.31
	MetaGCD [18]	56.20	79.59	39.83	56.63	65.81	55.05	53.65	62.76	52.05	55.49	69.38	48.98
	PA-CGCD [7]	57.43	80.29	41.43	61.69	92.38	55.84	55.63	88.67	49.85	58.25	87.11	49.04
	PromptCCD *w*/GMP (Ours)	70.69	80.90	63.54	64.08	73.14	62.35	57.73	72.67	55.12	**64.17**	75.57	60.34
ImageNet-100	ORCA [1]	47.63	69.84	32.09	34.46	38.95	33.60	38.77	28.76	40.52	40.29	45.85	35.40
	GCD [13]	75.65	84.69	69.31	71.21	80.67	69.40	61.38	79.33	58.23	69.41	81.56	65.65
	SimGCD [17]	36.96	51.43	28.29	32.46	40.71	30.36	24.73	29.29	23.67	31.38	40.47	27.44
	GCD *w*/replay	79.46	84.78	75.74	72.64	81.81	70.89	64.01	84.67	60.40	72.04	83.75	69.01
	SimGCD *w*/replay	48.76	77.14	31.71	48.55	67.14	43.82	45.27	59.29	42.00	47.53	67.86	39.18
	Grow & Merge [19]	75.34	76.78	74.34	63.76	73.67	62.87	64.43	74.86	62.60	67.84	75.10	66.60
	MetaGCD [18]	65.61	83.92	52.80	67.36	83.14	64.35	66.26	74.57	64.80	66.41	80.54	60.65
	PA-CGCD [7]	70.05	82.61	61.26	68.34	96.57	62.95	55.99	94.29	49.28	64.79	91.15	57.83
	PromptCCD *w*/GMP (Ours)	79.56	84.24	76.29	78.58	79.71	78.36	70.33	81.33	68.40	**76.16**	81.76	74.35
TinyImageNet	ORCA [1]	62.64	68.63	58.44	50.76	61.38	48.25	50.72	59.38	49.11	54.71	63.13	51.93
	GCD [13]	63.62	73.14	56.96	51.08	64.19	48.58	50.91	60.29	49.28	55.20	65.87	51.61
	SimGCD [17]	37.96	34.76	40.20	32.18	26.62	33.24	30.05	25.95	30.77	33.40	29.11	34.74
	GCD *w*/replay	66.40	76.06	59.64	50.28	66.38	47.21	52.32	60.19	50.94	56.33	67.54	52.60
	SimGCD *w*/replay	46.04	65.22	32.61	34.21	56.00	30.05	32.11	53.24	28.42	37.45	58.15	30.36
	Grow & Merge [19]	59.52	64.24	56.21	51.50	58.19	50.23	45.39	56.62	43.43	52.14	59.68	49.96
	MetaGCD [18]	59.41	73.90	49.27	57.21	63.71	55.96	49.17	60.76	47.14	55.26	66.12	50.79
	PA-CGCD [7]	56.01	74.96	42.74	41.89	65.38	37.40	55.50	84.52	50.42	51.13	74.95	43.52
	PromptCCD *w*/GMP (Ours)	68.67	72.84	65.76	59.69	65.67	58.55	57.16	61.10	56.47	**61.84**	66.54	60.26
Caltech-101	ORCA [1]	80.79	80.08	75.52	77.34	85.96	74.27	72.18	82.38	69.82	76.77	82.80	73.20
	GCD [13]	82.17	93.94	70.49	74.73	85.38	70.95	77.91	80.48	77.31	78.27	86.60	72.92
	SimGCD [17]	34.57	38.29	30.87	34.00	38.60	32.37	32.38	35.71	31.61	33.65	37.53	31.62
	GCD *w*/replay	87.11	92.56	81.69	75.50	85.38	71.99	66.91	80.48	63.77	76.51	86.14	72.48
	SimGCD *w*/replay	59.95	62.26	57.65	41.50	42.11	41.29	46.69	53.81	45.04	49.38	52.72	47.99
	Grow & Merge [19]	80.80	88.15	73.50	75.96	87.13	71.99	70.48	75.71	69.27	75.75	83.66	71.59
	MetaGCD [18]	86.15	95.59	76.78	75.96	90.06	70.95	80.14	81.43	79.85	80.75	89.02	75.86
	PA-CGCD [7]	78.33	92.56	64.21	79.33	98.83	72.41	76.21	92.86	72.36	77.96	94.75	69.66
	PromptCCD *w*/GMP (Ours)	89.57	92.84	86.34	78.87	87.72	75.73	78.89	86.67	77.09	**82.44**	89.08	79.72

Table B: Breakdown results of different methods for CCD leveraging pretrained DINOv2 model on generic datasets with the *known C* in each unlabelled set.

	Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	GCD [13]	74.18	90.16	63.00	63.89	73.52	62.05	57.99	67.52	56.32	65.35	77.06	60.46
	MetaGCD [18]	62.03	84.94	46.00	51.34	78.95	46.07	42.94	75.05	37.32	52.10	79.64	43.13
	PA-CGCD [7]	59.75	84.04	42.74	47.60	69.33	43.45	55.73	84.19	50.75	54.36	79.19	45.65
	PromptCCD *w*/GMP (Ours)	78.24	90.04	69.97	65.27	74.67	63.46	65.69	69.33	65.05	**69.73**	78.01	66.16
IN-100	GCD [13]	78.72	86.49	73.29	73.92	81.90	72.02	62.09	80.67	58.83	71.58	83.02	68.05
	MetaGCD [18]	71.56	86.49	61.11	70.14	80.14	68.75	68.90	81.24	64.12	70.20	82.62	64.66
	PA-CGCD [7]	79.83	87.92	74.17	67.53	86.29	63.95	77.11	90.38	77.93	74.82	88.20	72.02
	PromptCCD *w*/GMP (Ours)	80.35	86.78	75.86	75.65	80.86	74.65	72.85	80.19	73.08	**76.28**	82.61	74.53
Tiny	GCD [13]	69.68	82.90	60.43	56.57	77.81	52.52	50.89	71.62	47.27	59.05	77.44	53.41
	MetaGCD [18]	62.07	81.69	48.33	53.29	73.29	49.47	53.10	69.10	50.30	56.15	74.69	49.37
	PA-CGCD [7]	56.94	73.41	45.41	47.30	66.33	43.66	52.07	64.48	49.90	52.10	68.07	46.32
	PromptCCD *w*/GMP (Ours)	74.30	83.69	67.73	67.00	75.86	65.31	63.31	67.14	62.64	**68.20**	75.56	65.23
C-101	GCD [13]	89.99	95.87	84.15	79.63	91.81	75.31	79.38	78.27	79.93	83.00	88.65	79.80
	MetaGCD [18]	89.03	91.18	86.89	77.95	88.89	74.07	82.17	84.19	81.70	83.05	88.08	80.89
	PA-CGCD [7]	80.52	91.18	69.95	88.97	95.32	86.72	79.70	95.71	75.99	83.06	94.07	77.55
	PromptCCD *w*/GMP (Ours)	90.53	94.21	86.89	76.26	90.06	71.37	84.79	79.52	86.01	**83.86**	87.93	81.42

Table C: Breakdown results of different methods for CCD leveraging pretrained DINO model on fine-grained datasets with the *known C* in each unlabelled set.

	Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
Aircraft	ORCA [1]	31.89	31.43	32.17	28.64	27.14	28.93	31.77	18.57	36.23	30.77	25.71	32.44
	GCD [13]	47.30	58.57	40.43	48.59	68.57	44.66	46.21	57.41	42.51	47.37	61.43	42.53
	SimGCD [17]	32.97	47.86	23.91	30.28	34.29	29.49	23.83	25.00	23.43	29.03	35.72	25.61
	GCD *w*/replay	47.03	60.71	38.70	46.71	71.43	41.85	43.14	55.00	39.13	45.63	62.38	39.89
	SimGCD *w*/replay	41.08	62.86	27.83	33.33	67.14	26.69	37.91	54.29	32.37	37.44	61.43	28.96
	Grow & Merge [19]	32.43	32.86	32.17	30.05	41.43	27.81	30.69	25.71	32.37	31.06	33.33	30.78
	MetaGCD [18]	47.57	61.43	39.13	45.54	64.29	41.85	40.79	51.43	37.20	44.63	59.05	39.39
	PA-CGCD [7]	48.11	61.43	40.00	45.54	87.14	37.36	51.08	70.71	44.44	48.24	73.09	40.60
	PromptCCD *w*/GMP (Ours)	57.30	63.57	53.48	47.18	64.29	43.82	53.43	53.57	53.28	**52.64**	60.48	50.23
SCars	ORCA [1]	22.24	30.35	17.35	21.50	43.61	18.56	18.64	26.25	16.89	20.79	33.40	17.60
	GCD [13]	43.78	60.70	33.55	38.58	61.65	35.51	35.26	52.51	31.29	39.21	58.29	33.45
	SimGCD [17]	23.47	31.17	18.82	21.06	52.63	16.85	18.50	39.00	13.78	21.01	40.93	16.48
	GCD *w*/replay	43.67	62.60	32.24	40.18	58.65	37.71	35.77	53.28	31.73	39.87	58.18	33.89
	SimGCD *w*/replay	35.82	36.86	19.15	22.30	63.16	16.85	20.16	47.10	13.96	22.76	49.04	16.65
	Grow & Merge [19]	24.18	34.42	18.00	22.74	42.11	20.16	18.79	29.34	16.36	21.90	35.29	18.17
	MetaGCD [18]	39.80	56.37	29.79	34.25	61.65	30.59	33.89	52.90	29.51	35.98	56.97	29.96
	PA-CGCD [7]	49.31	75.88	33.86	39.62	74.69	35.11	42.70	90.73	31.64	43.88	80.43	33.54
	PromptCCD *w*/GMP (Ours)	50.31	71.82	37.32	44.69	62.41	42.33	37.21	64.86	30.84	**44.07**	66.36	36.83
CUB	ORCA [1]	49.79	66.43	38.66	30.63	63.57	23.64	44.76	68.57	40.11	41.73	66.19	34.14
	GCD [13]	58.80	75.71	47.49	49.25	76.43	43.48	56.88	74.49	53.48	54.98	75.47	48.15
	SimGCD [17]	49.26	63.81	39.71	29.48	49.64	25.23	40.92	64.29	36.30	39.89	59.25	33.75
	GCD *w*/replay	59.66	77.50	47.73	50.12	73.57	45.15	54.20	72.86	50.56	54.66	74.64	47.81
	SimGCD *w*/replay	49.70	75.75	32.60	37.99	69.34	31.38	38.56	72.86	31.78	42.08	72.65	31.92
	Grow & Merge [19]	44.21	65.00	30.31	29.50	65.00	21.97	42.89	65.00	38.58	38.87	65.00	30.29
	MetaGCD [18]	50.93	71.07	37.47	39.37	74.29	31.97	43.47	77.86	36.77	44.59	74.40	35.40
	PA-CGCD [7]	55.94	73.21	44.39	46.25	77.86	39.55	55.24	80.71	50.28	52.48	77.26	44.74
	PromptCCD *w*/GMP (Ours)	57.08	75.00	45.11	47.38	75.00	41.52	61.89	76.43	59.05	**55.45**	75.48	48.56

Table D: Breakdown results of different methods for CCD leveraging pretrained DINOv2 model on fine-grained datasets with the *known C* in each unlabelled set.

	Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
Aircraft	GCD [13]	60.00	70.71	53.48	56.57	67.14	54.49	57.04	53.57	58.21	57.87	63.80	55.39
Aircraft	MetaGCD [18]	58.11	64.29	54.35	55.16	74.29	51.40	51.44	54.29	50.48	54.90	64.29	52.08
Aircraft	PA-CGCD [7]	59.73	75.71	50.00	52.82	72.86	48.88	61.91	84.29	54.35	58.15	77.62	51.08
Aircraft	PromptCCD *w*/GMP (Ours)	61.35	68.57	56.96	58.92	72.86	56.18	67.87	63.57	69.32	**62.71**	68.33	60.82
SCars	GCD [13]	60.51	77.51	50.25	58.41	71.43	56.67	56.65	66.02	54.49	58.52	71.65	53.80
SCars	MetaGCD [18]	59.08	76.15	48.77	57.35	68.42	55.87	55.06	71.04	51.38	57.16	71.87	52.01
SCars	PA-CGCD [7]	68.51	85.80	58.07	65.08	88.98	61.89	61.15	94.14	53.56	64.91	89.64	57.84
SCars	PromptCCD *w*/GMP (Ours)	69.49	85.09	60.07	65.49	73.68	64.39	60.26	71.04	57.78	**65.08**	76.60	60.75
CUB	GCD [13]	68.67	85.00	57.76	66.87	80.71	63.94	64.57	84.29	60.72	66.70	83.33	60.81
CUB	MetaGCD [18]	65.81	84.64	53.22	57.25	80.00	52.42	63.52	82.86	59.75	62.19	82.50	55.13
CUB	PA-CGCD [7]	67.67	87.86	54.18	65.25	99.29	58.03	67.72	90.71	63.23	66.88	92.62	58.48
CUB	PromptCCD *w*/GMP (Ours)	69.10	83.21	59.67	63.00	80.00	59.39	71.33	81.43	69.36	**67.81**	81.55	62.81

Table E: Breakdown results of our method with different prompt pool designs for CCD leveraging pretrained DINO model on generic and CUB datasets with the *known C* in each unlabelled set. The experiments are conducted with *seed 1*.

	Method	Prompt Pool	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
			All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	PromptCCD-B (Ours)	L2P [16]	65.93	80.20	55.94	56.72	70.76	54.04	51.55	66.67	48.90	58.07	72.54	52.96
C100	PromptCCD-B (Ours)	DP [15]	69.92	82.57	61.06	55.21	78.10	51.75	54.37	74.29	50.88	59.83	78.32	54.56
C100	PromptCCD (Ours)	GMP (Ours)	70.69	80.90	63.54	64.08	73.14	62.35	57.73	72.67	55.12	**64.17**	75.57	60.34
IN-100	PromptCCD-B (Ours)	L2P [16]	75.80	83.84	70.17	70.95	82.48	68.75	61.19	78.67	58.13	69.31	81.66	65.68
IN-100	PromptCCD-B (Ours)	DP [15]	76.39	83.43	71.46	67.90	83.52	63.96	61.13	83.24	57.57	68.47	83.40	64.33
IN-100	PromptCCD (Ours)	GMP (Ours)	79.56	84.24	76.29	78.58	79.71	78.36	70.33	81.33	68.40	**76.16**	81.76	74.35
Tiny	PromptCCD-B (Ours)	L2P [16]	69.46	75.24	65.41	54.64	65.43	52.58	44.88	59.43	42.33	56.33	66.70	53.44
Tiny	PromptCCD-B (Ours)	DP [15]	65.91	72.63	61.60	55.73	67.71	53.44	48.28	59.14	46.37	56.64	66.49	53.80
Tiny	PromptCCD (Ours)	GMP (Ours)	68.67	72.84	65.76	59.69	65.67	58.55	57.16	61.10	56.47	**61.84**	66.54	60.26
CUB	PromptCCD-B (Ours)	L2P [16]	56.65	73.93	45.11	47.75	74.29	42.12	47.32	71.43	42.62	50.57	73.22	43.28
CUB	PromptCCD-B (Ours)	DP [15]	60.66	77.86	49.16	47.62	74.57	41.91	57.34	79.29	53.06	55.21	77.24	48.04
CUB	PromptCCD (Ours)	GMP (Ours)	57.08	75.00	45.11	47.38	75.00	41.52	61.89	76.43	59.05	**55.45**	75.48	48.56

Table F: Breakdown results of our method with different prompt pool designs for CCD leveraging pretrained DINO model on generic and CUB datasets with the *known C* in each unlabelled set. The experiments are conducted with *seed 7*.

	Method	Prompt Pool	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
			All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	PromptCCD-B (Ours)	L2P [16]	46.54	61.06	36.37	38.95	48.57	37.11	38.48	46.29	37.12	41.32	51.97	36.87
C100	PromptCCD-B (Ours)	DP [15]	69.92	82.57	61.06	56.90	78.10	52.85	48.14	70.67	44.20	58.32	77.11	52.70
C100	PromptCCD (Ours)	GMP (Ours)	74.54	82.45	69.00	65.82	77.81	63.53	58.07	83.62	53.60	**66.14**	81.29	62.04
IN-100	PromptCCD-B (Ours)	L2P [16]	70.45	80.37	63.51	65.80	80.38	63.02	58.40	77.05	55.13	64.88	79.27	60.55
IN-100	PromptCCD-B (Ours)	DP [15]	80.12	83.31	77.89	69.33	82.76	66.76	65.94	84.00	62.78	71.80	83.36	69.14
IN-100	PromptCCD (Ours)	GMP (Ours)	78.40	81.96	75.91	76.66	80.57	75.91	67.60	78.76	65.65	**74.22**	80.43	72.49
Tiny	PromptCCD-B (Ours)	L2P [16]	64.83	73.02	59.10	55.63	65.76	53.69	51.15	57.43	50.05	57.20	65.40	54.28
Tiny	PromptCCD-B (Ours)	DP [15]	66.72	74.51	61.27	58.00	66.52	56.37	55.45	61.38	54.42	60.06	67.47	57.35
Tiny	PromptCCD (Ours)	GMP (Ours)	68.31	73.41	64.74	58.36	66.76	56.75	55.45	61.38	54.42	**60.71**	63.85	58.64
CUB	PromptCCD-B (Ours)	L2P [16]	55.36	74.29	42.72	43.00	75.00	36.21	55.24	71.43	52.09	51.20	73.57	43.67
CUB	PromptCCD-B (Ours)	DP [15]	55.94	77.50	41.53	47.75	80.00	40.91	63.75	77.14	61.14	55.81	78.21	47.86
CUB	PromptCCD (Ours)	GMP (Ours)	58.37	78.21	45.11	50.25	77.14	44.55	61.54	85.71	56.82	**56.72**	80.35	48.83

Table G: Breakdown results of our method with different prompt pool designs for CCD leveraging pretrained DINO model on generic and CUB datasets with the *known* C in each unlabelled set. The experiments are conducted with *seed 10*.

	Method	Prompt Pool	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
			All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	PromptCCD-B (Ours)	L2P [16]	64.32	78.04	54.71	52.58	69.62	49.33	46.79	60.95	44.32	54.56	69.54	49.45
	PromptCCD-B (Ours)	DP [15]	71.82	83.51	63.63	58.44	77.90	54.73	49.22	75.90	44.55	59.83	79.10	54.30
	PromptCCD (Ours)	GMP (Ours)	73.82	79.67	69.71	63.48	74.00	61.47	55.46	72.67	52.45	**64.25**	75.45	61.21
IN-100	PromptCCD-B (Ours)	L2P [16]	69.09	83.80	58.80	65.68	81.14	62.73	63.50	79.81	60.65	66.09	81.58	60.73
	PromptCCD-B (Ours)	DP [15]	78.08	84.12	73.86	72.89	83.24	70.91	63.05	85.14	59.18	71.34	84.17	67.98
	PromptCCD (Ours)	GMP (Ours)	79.97	84.41	76.86	77.16	79.71	76.67	69.39	78.00	67.88	**75.71**	80.71	73.80
Tiny	PromptCCD-B (Ours)	L2P [16]	67.50	75.53	61.87	53.47	64.81	51.31	49.73	59.05	48.10	56.90	66.46	53.76
	PromptCCD-B (Ours)	DP [15]	66.93	74.94	61.33	58.53	63.43	57.60	55.09	59.90	54.24	60.18	66.09	57.72
	PromptCCD (Ours)	GMP (Ours)	69.95	78.51	63.51	59.22	65.67	56.04	57.88	63.67	54.88	**62.25**	69.28	58.14
CUB	PromptCCD-B (Ours)	L2P [16]	53.65	70.00	42.72	49.00	72.86	43.94	55.48	71.43	52.37	52.71	71.43	46.34
	PromptCCD-B (Ours)	DP [15]	61.80	80.36	49.40	49.38	80.71	42.73	58.74	80.00	54.60	56.64	80.36	48.91
	PromptCCD (Ours)	GMP (Ours)	61.09	81.07	47.73	50.12	85.71	42.58	60.26	79.29	56.55	**57.16**	82.02	48.95

Table H: Breakdown results of our method with different prompt pool designs for CCD leveraging pretrained DINO model on generic and CUB datasets with the *known* C in each unlabelled set. The experiments are conducted with *seed 2000*.

	Method	Prompt Pool	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
			All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	PromptCCD-B (Ours)	L2P [16]	60.40	76.61	49.06	50.79	65.62	47.20	41.04	65.43	36.77	50.74	69.22	44.34
	PromptCCD-B (Ours)	DP [15]	70.97	83.67	62.09	60.33	77.81	57.13	44.52	78.29	38.62	58.61	79.92	52.61
	PromptCCD (Ours)	GMP (Ours)	70.00	80.94	62.34	64.66	74.10	62.85	53.79	72.19	50.57	**62.82**	75.74	58.59
IN-100	PromptCCD-B (Ours)	L2P [16]	74.97	83.35	69.11	70.08	81.52	67.89	56.18	84.00	51.32	67.08	82.96	62.77
	PromptCCD-B (Ours)	DP [15]	77.51	83.96	73.00	70.21	82.29	67.91	65.65	83.71	62.48	71.12	83.32	67.80
	PromptCCD (Ours)	GMP (Ours)	80.30	83.10	78.34	74.92	82.29	73.51	70.54	80.48	68.80	**75.25**	81.96	73.55
Tiny	PromptCCD-B (Ours)	L2P [16]	66.36	73.41	61.43	54.65	66.38	52.41	49.02	60.33	47.04	56.68	66.71	53.63
	PromptCCD-B (Ours)	DP [15]	66.89	73.98	61.93	55.05	66.90	52.78	50.68	60.24	49.01	57.54	67.04	54.57
	PromptCCD (Ours)	GMP (Ours)	66.82	72.16	63.07	59.69	66.57	58.38	57.26	60.33	56.73	**61.26**	66.35	59.39
CUB	PromptCCD-B (Ours)	L2P [16]	52.65	70.36	40.81	46.50	74.29	40.61	51.52	70.00	47.91	50.22	71.55	43.11
	PromptCCD-B (Ours)	DP [15]	61.95	80.00	49.88	51.25	80.71	45.00	61.19	80.71	57.38	**58.13**	80.47	50.75
	PromptCCD (Ours)	GMP (Ours)	56.80	75.00	44.63	48.88	78.57	42.58	62.00	86.43	57.24	55.89	80.00	48.15

Table I: Breakdown results of our method with different prompt pool designs for CCD leveraging pretrained DINO model on generic and CUB datasets with the *known* C in each unlabelled set. The experiments are conducted with *seed 2024*.

	Method	Prompt Pool	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
			All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	PromptCCD-B (Ours)	L2P [16]	65.56	80.53	55.09	52.26	69.33	49.00	41.87	69.33	37.07	53.23	73.06	47.05
	PromptCCD-B (Ours)	DP [15]	74.79	83.47	68.71	60.34	77.71	57.02	49.18	79.52	43.87	61.44	80.23	56.53
	PromptCCD (Ours)	GMP (Ours)	69.48	82.57	60.31	62.96	73.71	60.91	55.02	69.62	52.47	**62.49**	75.30	57.90
IN-100	PromptCCD-B (Ours)	L2P [16]	68.87	82.82	57.51	64.19	79.24	60.36	56.92	77.33	53.30	63.33	79.80	57.06
	PromptCCD-B (Ours)	DP [15]	76.82	84.16	71.69	71.74	82.95	69.60	62.89	82.10	59.53	70.48	83.07	66.94
	PromptCCD (Ours)	GMP (Ours)	80.64	83.35	78.64	79.24	80.86	78.93	67.45	78.67	65.48	**75.78**	80.96	74.35
Tiny	PromptCCD-B (Ours)	L2P [16]	66.03	73.39	60.89	52.70	62.38	50.85	49.88	59.10	48.27	56.20	64.96	53.34
	PromptCCD-B (Ours)	DP [15]	63.58	72.51	57.33	56.19	64.43	54.62	56.15	60.95	55.31	58.64	65.96	55.75
	PromptCCD (Ours)	GMP (Ours)	66.88	72.08	63.24	57.23	63.33	56.06	54.91	60.90	53.87	**59.67**	65.44	57.72
CUB	PromptCCD-B (Ours)	L2P [16]	52.50	70.00	40.81	47.25	76.43	41.06	55.83	70.71	52.92	51.86	72.38	44.93
	PromptCCD-B (Ours)	DP [15]	58.66	74.29	48.21	50.00	78.57	43.94	58.51	77.86	54.74	55.72	76.91	48.96
	PromptCCD (Ours)	GMP (Ours)	61.37	80.36	48.69	51.75	78.57	46.06	60.96	85.71	56.13	**58.03**	81.55	50.29

Table J: Breakdown results of different methods for CCD leveraging pretrained DINO model on generic and CUB datasets with the *unknown* C in each unlabelled set. The Cs are estimated using our method described in Sec. 3.3 , in the main paper. The estimated Cs are applied to all other methods for comparison.

	Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	Estimated C	(C^{EST}: 77, C^{GT}: 80)			(C^{EST}: 78, C^{GT}: 90)			(C^{EST}: 81, C^{GT}: 100)					
	GCD [13]	67.13	80.49	57.77	51.51	63.43	49.24	52.96	60.76	51.60	57.20	68.23	52.87
	Grow & Merge [19]	58.15	68.82	50.69	56.95	58.57	56.64	54.68	54.38	54.73	56.59	60.59	54.02
	MetaGCD [18]	57.45	78.65	42.60	40.40	61.62	36.35	61.74	58.23	62.33	53.20	66.22	47.09
	PA-CGCD [7]	57.46	80.37	41.43	55.16	84.10	49.64	56.75	84.10	51.97	56.46	82.86	47.68
	PromptCCD-U w/GMP (Ours)	69.09	79.76	61.43	62.29	68.00	61.20	58.21	65.81	56.88	**63.20**	71.19	59.90
IN-100	Estimated C	(C^{EST}: 73, C^{GT} : 80)			(C^{EST}: 73, C^{GT} : 90)			(C^{EST}: 83, C^{GT} : 100)					
	GCD [13]	56.44	82.29	38.34	53.79	72.10	50.29	59.57	78.86	56.20	56.60	77.75	48.28
	Grow & Merge [19]	53.29	75.02	38.09	41.47	72.00	35.64	63.74	70.00	62.65	52.83	72.34	45.46
	MetaGCD [18]	51.43	82.53	29.66	38.81	72.00	32.47	63.97	72.67	62.45	51.40	75.73	41.53
	PA-CGCD [7]	51.95	81.14	31.51	35.66	85.52	26.15	58.28	86.86	53.28	48.63	84.51	36.98
	PromptCCD-U w/GMP (Ours)	60.49	82.61	45.00	64.63	71.33	63.35	64.31	76.10	62.25	**63.14**	76.68	56.87
Tiny	Estimated C	(C^{EST}: 158, C^{GT} : 160)			(C^{EST}: 164, C^{GT} : 180)			(C^{EST}: 168, C^{GT} : 200)					
	GCD [13]	62.80	72.37	56.10	49.63	62.81	47.11	48.61	58.95	46.80	53.68	64.71	50.00
	Grow & Merge [19]	57.19	62.61	53.40	50.40	54.38	49.65	52.72	52.48	52.77	53.44	56.49	51.94
	MetaGCD [18]	59.16	72.94	49.51	58.04	59.81	57.70	56.55	55.71	56.70	57.92	62.82	54.64
	PA-CGCD [7]	56.87	73.84	44.99	43.82	61.33	40.47	50.50	75.33	46.15	50.40	70.17	43.87
	PromptCCD-U w/GMP (Ours)	67.32	79.63	58.70	60.77	70.52	58.91	54.35	63.05	52.82	**60.81**	71.07	56.81
CUB	Estimated C	(C^{EST}: 163, C^{GT}: 160)			(C^{EST}: 172, C^{GT}: 180)			(C^{EST}: 175, C^{GT}: 200)					
	GCD [13]	57.80	76.07	45.58	44.00	71.43	38.18	48.25	66.43	44.71	50.02	71.31	42.82
	Grow & Merge [19]	42.49	63.57	28.40	27.25	63.57	19.55	40.91	64.29	36.35	36.88	63.81	28.10
	MetaGCD [18]	54.36	72.14	42.48	38.75	70.00	32.12	40.56	68.57	35.10	44.56	70.24	36.57
	PA-CGCD [7]	58.80	76.43	47.02	46.38	70.86	39.91	46.65	76.29	40.87	50.61	74.53	42.60
	PromptCCD-U w/GMP (Ours)	56.94	77.14	43.44	47.25	72.14	41.97	49.41	70.71	45.11	**51.20**	73.33	43.51

Table K: Breakdown results of different methods for CCD leveraging pretrained DINO model on generic and CUB datasets with the *unknown* C in each unlabelled set. The Cs are estimated using the k-means-based estimator in [13]. The estimated Cs are applied to all other methods for comparison.

	Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	Estimated C	(C^{EST}: 84, C^{GT}: 80)			(C^{EST}: 84, C^{GT}: 90)			(C^{EST}: 84, C^{GT}: 100)					
	GCD [13]	62.45	83.27	47.89	52.23	63.71	50.04	49.32	58.86	47.65	54.67	68.61	48.53
	Grow & Merge [19]	63.45	72.29	57.26	57.51	57.52	57.51	54.90	51.05	55.10	58.62	60.29	56.62
	MetaGCD [18]	54.59	81.18	35.97	40.85	62.00	36.82	61.91	57.52	62.68	52.45	66.90	45.16
	PA-CGCD [7]	55.29	82.49	36.26	59.33	88.29	53.80	53.73	82.76	48.65	56.12	84.51	46.24
	PromptCCD w/GMP (Ours)	69.38	82.78	60.00	64.55	70.67	63.38	59.23	64.76	58.27	**64.39**	72.74	60.55
IN-100	Estimated C	(C^{EST}: 90, C^{GT}: 80)			(C^{EST}: 90, C^{GT}: 90)			(C^{EST}: 91, C^{GT}: 100)					
	GCD [13]	64.66	84.53	50.74	71.54	76.57	70.58	56.67	74.67	53.52	64.29	78.59	58.28
	Grow & Merge [19]	65.19	76.29	57.43	58.56	70.95	56.20	55.06	71.71	52.15	59.60	72.98	55.26
	MetaGCD [18]	55.58	83.88	35.77	45.76	77.90	39.62	60.00	73.43	57.65	53.78	78.40	44.35
	PA-CGCD [7]	58.03	82.49	40.91	51.45	89.62	44.16	51.02	86.57	44.80	53.50	86.23	43.29
	PromptCCD w/GMP (Ours)	70.77	84.12	61.43	72.03	75.71	71.33	59.90	77.71	56.78	**67.57**	79.18	63.18
Tiny	Estimated C	(C^{EST}: 169, C^{GT}: 160)			(C^{EST}: 169, C^{GT}: 180)			(C^{EST}: 172, C^{GT}: 200)					
	GCD [13]	65.45	73.06	60.11	50.78	61.05	48.82	47.21	55.52	45.75	54.48	63.21	51.56
	Grow & Merge [19]	56.78	64.12	51.64	49.03	54.14	48.05	52.39	53.48	52.20	52.73	57.25	50.63
	MetaGCD [18]	59.76	74.22	49.63	58.46	58.38	58.47	60.34	56.95	60.93	59.52	63.18	56.34
	PA-CGCD [7]	53.80	74.65	39.20	41.19	62.76	37.07	51.72	76.33	47.42	48.90	71.25	41.23
	PromptCCD w/GMP (Ours)	66.73	79.76	57.61	57.49	68.38	55.41	56.35	65.57	54.74	**60.19**	71.24	55.92
CUB	Estimated C	(C^{EST}: 166, C^{GT}: 160)			(C^{EST}: 192, C^{GT}: 180)			(C^{EST}: 220, C^{GT}: 200)					
	GCD [13]	58.51	77.50	45.82	51.00	75.71	45.76	53.03	78.57	48.05	54.18	77.26	46.54
	Grow & Merge [19]	43.20	62.50	30.31	31.62	67.14	24.09	43.12	65.71	38.72	39.31	65.12	31.04
	MetaGCD [18]	52.50	71.07	40.10	43.50	77.14	36.36	46.62	70.00	42.06	47.54	72.74	39.51
	PA-CGCD [7]	58.23	74.29	47.49	51.50	78.57	45.76	56.06	77.14	51.95	55.26	76.67	48.40
	PromptCCD w/GMP (Ours)	59.94	80.00	46.54	52.50	78.57	46.97	54.20	76.43	49.86	**55.55**	78.33	47.79

S3 Transductive and Inductive Evaluation

In our main paper, we evaluate our method on the unlabelled data, which are from the train splits of the original datasets. Indeed, the model has seen the data during training, though no labels are used. Here, we further evaluate our method on the test splits of the original datasets, which were not seen by the model during training. In other words, we consider two evaluation protocols, namely, *transductive evaluation* and *inductive evaluation*. In *transductive evaluation*, the model is evaluated on the unlabelled data that has been seen by the model during training, while in *inductive evaluation*, the model is evaluated on the unlabelled data that has not been seen by the model during training.

Since we have reported the transductive evaluation results in the main paper, here, we further include the inductive evaluation results in Tab. L, based on the *cACC* evaluation metric introduced in the main paper. Overall, we can see that our method is more robust to unseen data compared to other models as it consistently performs better in the *'All'* and *'New'* accuracy.

Table L: Comparison using the *cACC* evaluation metric under the *inductive* protocol.

	Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	GCD [13]	64.37	86.33	53.29	53.89	70.00	50.82	51.21	65.71	48.67	56.49	74.01	50.93
	Grow & Merge [19]	56.05	66.53	48.71	58.17	65.71	56.73	53.33	63.33	51.58	55.85	65.19	52.34
	MetaGCD [18]	56.47	78.37	41.14	41.22	67.14	36.27	56.88	63.81	55.67	51.52	69.77	44.36
	PA-CGCD [7]	57.14	79.80	41.29	53.05	74.76	48.91	53.90	68.57	51.33	54.70	**74.38**	47.18
	PromptCCD *w*/GMP (Ours)	67.98	79.39	60.00	60.31	74.76	57.55	57.02	65.71	55.50	**61.77**	73.29	**57.68**
IN-100	GCD [13]	74.11	78.57	71.43	71.88	77.86	70.36	60.95	77.86	57.00	68.98	78.10	66.26
	Grow & Merge [19]	77.50	77.14	77.71	73.04	75.71	72.36	63.38	73.57	61.00	71.31	75.47	70.36
	MetaGCD [18]	66.43	84.76	55.43	61.01	78.57	56.55	68.51	71.43	67.83	65.32	78.25	59.94
	PA-CGCD [7]	64.64	81.43	54.57	58.12	72.86	54.36	60.81	77.14	57.00	61.19	77.14	55.31
	PromptCCD *w*/GMP (Ours)	78.75	79.52	78.29	75.80	78.57	75.09	69.46	80.00	67.00	**74.67**	**79.36**	**73.46**
Tiny	GCD [13]	60.45	73.81	52.43	46.09	61.79	42.09	46.49	60.71	43.17	51.01	65.44	45.90
	Grow & Merge [19]	56.52	65.00	51.43	38.84	55.71	34.55	47.30	56.07	45.25	47.55	58.93	43.74
	MetaGCD [18]	50.98	70.71	39.14	49.71	60.00	47.09	51.82	60.71	49.75	50.84	63.81	45.33
	PA-CGCD [7]	51.79	70.71	40.43	40.07	64.64	33.82	47.97	64.29	44.17	46.61	**66.55**	39.47
	PromptCCD *w*/GMP (Ours)	61.96	70.24	57.00	52.46	64.64	49.36	49.86	62.50	46.92	**53.76**	65.79	**51.09**
CUB	GCD [13]	62.72	78.73	52.21	50.95	75.71	45.69	49.17	78.57	43.36	54.28	77.67	47.09
	Grow & Merge [19]	45.86	64.93	33.33	29.35	60.58	22.77	41.51	62.86	37.29	38.91	62.79	31.13
	MetaGCD [18]	54.44	72.39	42.65	45.36	78.10	38.46	50.47	70.71	46.47	50.09	73.73	42.53
	PA-CGCD [7]	57.69	81.34	42.16	50.57	73.72	45.69	58.85	78.57	56.21	55.70	77.88	48.02
	PromptCCD *w*/GMP (Ours)	58.88	77.99	46.32	50.32	78.83	44.31	63.33	77.14	60.59	**57.51**	**77.99**	**50.41**

S4 Adapting the *ACC* Metric in GCD for CCD in Each Time Step

In the main paper, when evaluating our method, at each time step t, we consider the previously discovered categories as "known" (associated with the pseudo labels obtained by our method), which are included in D^l in the $cACC$ evaluation algorithm. Here, we additionally show the results of applying the commonly used ACC metric in GCD to each time step in CCD. Particularly, in each time step, we measure the ACC based on $D^l \cup D^u_t$. The ACC can be computed following [13]. We summarize the adapted evaluation metric in Alg. B and report the results in Tab. M. Overall, our method consistently outperforms other methods on all datasets on 'All' and 'New' splits.

Algorithm B standard incremental GCD evaluation metric

Input: Models $\{f^t_\theta \mid t = 1, \ldots, T\}$ and datasets $\{D^l, D^u\}$.
Output: ACC value.
Require: SS-k-MEANS(Model, Labelled set, Unlabelled set).
for $t \in \{1, \cdots, T\}$ **do**
  $ACC_t \leftarrow$ SS-k-MEANS(f^t_θ, D^l, D^u_t)
$ACCs \leftarrow \{ACC_t \mid t = 1, \ldots, T\}$
$ACC \leftarrow$ AVERAGE($ACCs$)
return ACC

Table M: Comparison using the adapted ACC metric from GCD in Alg. B.

	Method	Stage 1 ACC (%)			Stage 2 ACC (%)			Stage 3 ACC (%)			Average ACC (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	GCD [13]	67.65	83.59	56.49	49.85	71.52	45.71	44.51	81.62	38.02	54.00	78.91	46.74
	Grow & Merge [19]	64.77	70.49	60.77	61.27	64.00	60.75	43.56	59.90	40.70	56.53	64.80	54.07
	MetaGCD [18]	56.20	79.59	39.83	54.09	68.57	57.51	44.77	60.10	42.08	51.69	69.42	46.47
	PA-CGCD [7]	57.43	80.29	41.43	63.42	86.76	55.05	50.81	83.24	45.73	57.22	**83.43**	47.40
	PromptCCD w/GMP (Ours)	70.69	80.90	63.54	65.65	74.57	63.95	51.94	83.62	46.40	**62.76**	79.70	**57.96**
IN-100	GCD [13]	75.65	84.69	69.31	64.39	78.71	62.65	53.49	80.19	48.82	64.51	81.20	60.26
	Grow & Merge [19]	75.34	76.78	74.34	63.11	78.00	60.27	54.06	73.71	50.62	64.17	76.16	61.74
	MetaGCD [18]	65.61	83.92	52.80	60.23	85.05	55.49	65.09	77.43	62.93	63.64	82.13	57.07
	PA-CGCD [7]	70.05	82.61	61.26	66.40	95.52	60.84	62.41	93.81	56.92	66.29	**90.65**	59.67
	PromptCCD w/GMP (Ours)	79.56	84.24	76.29	66.34	81.43	63.45	58.91	78.95	64.05	**68.27**	81.54	**67.93**
Tiny	GCD [13]	63.62	73.14	56.96	54.09	67.19	51.59	47.98	62.48	45.44	55.23	67.60	51.33
	Grow & Merge [19]	59.52	64.24	56.21	50.19	57.95	48.71	52.06	54.90	51.57	53.92	59.03	52.16
	MetaGCD [18]	59.41	73.90	49.27	59.90	61.90	59.52	53.43	61.29	52.06	57.58	65.70	53.62
	PA-CGCD [7]	56.01	74.96	42.74	46.81	67.14	42.93	52.74	88.86	46.92	51.85	**76.99**	44.03
	PromptCCD w/GMP (Ours)	68.67	72.84	65.76	60.11	73.48	57.56	51.51	60.71	49.90	**60.10**	69.01	**57.74**
CUB	GCD [13]	58.80	75.71	47.49	47.50	80.71	40.45	47.67	76.43	42.06	51.32	77.62	43.33
	Grow & Merge [19]	44.21	65.00	30.31	32.25	69.29	24.39	37.18	67.14	31.34	37.88	67.14	28.68
	MetaGCD [18]	50.93	71.07	37.47	43.50	75.71	36.67	45.92	77.86	39.69	46.78	74.88	37.94
	PA-CGCD [7]	55.94	73.21	44.39	53.00	76.43	48.03	57.46	86.43	51.81	55.47	78.69	48.08
	PromptCCD w/GMP (Ours)	57.08	75.00	45.11	52.87	85.71	45.91	58.04	79.29	53.90	**56.00**	**80.00**	**48.31**

S5 Implementation Details for Augmenting Grow & Merge with ViT

As the most relevant work Grow & Merge (G&M) [19] uses ResNet18 [6] as the backbone and the Momentum Contrast (MoCo) [5] for representation learning, to have a fair comparison, we augment G&M from two aspects, the pretraining strategy and the dual branch network (static and dynamic branch), leveraging the more powerful ViT backbone. First, we change the pretraining strategy MoCo to joint supervised and unsupervised contrastive learning with DINO features. Second, for the dual branch network in [19], originally, the ResNet18 is divided into several layers (excluding the fully connected layers) where before the last layer, G&M divides the last layer into two branches, *i.e.*, the static branch and the dynamic branch. By design, the static branch is the backbone's last layer, while the dynamic branch consists of several branches of $T-1$ layers, where T is the number of stages. To maintain this design, we accordingly implement a dual-branch architecture network based on ViT backbone. Given that ViT backbone consists of several blocks, we freeze all blocks except the last block as the static branch. Moreover, before the last block, we add another $T-1$ blocks as the dynamic branches used exclusively for each stage t. All the rest designs are the same as [19]. At $t=0$, *i.e.*, during the initial stage, we optimize the static branch, and at $t>0$, we freeze the static branch and perform static-dynamic distillation while optimizing the dynamic branch t for novel class discovery following G&M.

We compare our method with the *improved* G&M under both *transductive* and *inductive* evaluation protocols, using the $cACC$ evaluation metric. As shown in Tab. N, our *improved* G&M significantly outperforms the original implementation, leading to a fair comparison with our method. However, our method obtains an overall accuracy of 64.17%, which is still substantially better.

Note: For experiments in the main paper, we compare our model with the *improved* Grow & Merge (G&M).

Table N: Comparison with different Grow & Merge implementations on CIFAR100 datasets.

Method	Stage 1 ACC (%) *All*	Stage 1 *Old*	Stage 1 *New*	Stage 2 ACC (%) *All*	Stage 2 *Old*	Stage 2 *New*	Stage 3 ACC (%) *All*	Stage 3 *Old*	Stage 3 *New*	Average ACC (%) *All*	Average *Old*	Average *New*
Transductive Evaluation												
Grow & Merge [19]	22.91	30.20	17.80	21.47	25.71	20.65	24.91	24.00	27.25	23.10	26.64	21.90
Grow & Merge (*improved*)	64.77	70.49	60.77	58.32	62.95	57.44	49.21	57.62	47.73	**57.43**	**63.68**	**55.31**
PromptCCD w/GMP (Ours)	70.69	80.90	63.54	64.08	73.14	62.35	57.73	72.67	55.12	**64.17**	**75.57**	**60.34**
Inductive Evaluation												
Grow & Merge [19]	38.32	60.61	22.71	29.62	60.48	23.73	31.91	60.95	26.83	33.28	60.68	24.42
Grow & Merge (*improved*)	64.77	70.49	60.77	61.27	64.00	60.75	43.56	59.90	40.70	**56.53**	**64.80**	**54.07**
PromptCCD w/GMP (Ours)	67.98	79.39	60.00	60.31	74.76	57.55	57.02	65.71	55.50	**61.77**	**73.29**	**57.68**

S6 Additional Comparison under Other CCD Settings and Metrics

In this section, we provide additional comparison with PA-CGCD [7] and MetaGCD [18], following their experimental settings, including data splits and evaluation protocols. In Tab. O, we follow PA-CGCD's experimental setting and use their evaluation metrics (described in Sec. 4.2 of [7]). As can be seen, our method achieves the best performance following the setting of [7]. In addition, we also follow MetaGCD's data splits. The common GCD evaluation metric, ACC, is adopted in their original paper. We experiment under their setting, and report the results in Tab. P. As can be seen, our method outperforms MetaGCD and all other methods. These results further demonstrate the superiority of our methods.

Table O: Comparison with PA-CGCD [7] on CUB. For experiment settings and evaluation metrics, please refer to the original paper's Sec. 4.2 (Tab. 4, DINO ViT-B/16 experiments).

Method	$\mathcal{M}_{all} \uparrow$	$\mathcal{M}_o \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$
GCD [13]	62.70	71.40	09.57	56.01
Grow & Merge [19]	42.12	60.21	23.24	27.63
PA-CGCD [7]	72.51	74.28	09.49	65.60
PromptCCD w/GMP (Ours)	**76.23**	**78.44**	**06.07**	**74.46**

Table P: Comparison with MetaGCD [18] on CIFAR100. For experimental settings and evaluation metric, please refer to the original paper's Sec. 4.

Method	Stage 1 ACC (%)			Stage 2 ACC (%)			Stage 3 ACC (%)			Stage 4 ACC (%)			Average ACC (%)		
	All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
RankStats [3]	62.33	64.22	31.60	55.01	58.55	26.85	51.77	56.70	25.47	47.51	54.59	17.20	54.16	58.52	25.28
FRoST [11]	67.14	68.57	50.73	67.01	68.82	52.60	62.35	65.48	45.67	55.84	59.06	42.95	63.09	65.48	47.99
GCD [13]	76.78	77.91	58.60	73.67	75.29	60.70	72.77	74.72	62.33	71.44	74.75	58.20	73.67	75.67	59.96
Grow & Merge [19]	78.29	79.91	66.00	77.58	79.64	61.13	74.56	77.60	58.14	72.02	75.98	56.32	75.61	78.28	60.40
MetaGCD [18]	78.96	79.36	72.60	78.67	79.41	66.81	76.06	78.20	64.87	74.56	77.60	61.14	77.06	78.64	66.35
PromptCCD w/GMP (Ours)	90.06	90.50	89.47	82.67	88.80	76.23	81.48	84.60	78.80	70.30	75.87	67.64	**81.13**	**84.94**	**78.04**

S7 Experiments on Additional Data Splits

The data splits in the main paper follow [19], as reported in Tab. 2 . The classes are split into 7:1:1:1, while the samples (D^l, D_1^u, D_2^u, D_3^u) in each stage are divided following the percentages in Tab. 2. To further mimic the real-world scenario, which is characterized by an abrupt increase or decrease in the number of classes of each stage, we experiment on another 3 different class splits: (1) 4:2:2:2 – the number of the unseen classes is greater than that of the seen classes; (2) 4:3:2:1 – the number of the unseen classes is decreasing for each stage; (3) 1:2:3:4 – the number of the unseen class is increasing for each stage. As shown in Tab. Q, we compare our model with GCD and Grow & Merge on the CIFAR100 dataset. Our model consistently outperforms others by a large margin across the board.

Table Q: Experiments on different class splits scenarios on CIFAR100.

Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
	All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
Class Split: 4:2:2:2												
GCD [13]	78.25	55.36	80.54	65.79	39.83	66.98	38.72	39.83	38.50	60.92	45.01	62.01
Grow & Merge [19]	51.17	41.86	52.10	45.90	31.50	46.57	34.72	45.17	32.63	43.93	39.51	43.77
PromptCCD *w*/GMP	78.53	50.50	81.34	74.22	59.67	74.89	52.64	43.50	54.47	**68.46**	**51.22**	**70.23**
Class Split: 4:3:2:1 (decreasing)												
GCD [13]	58.62	59.79	58.50	51.99	42.17	52.44	40.69	40.83	40.67	50.43	47.59	50.54
Grow & Merge [19]	41.89	52.83	39.70	44.25	42.83	44.32	34.97	35.50	34.87	40.37	43.72	39.63
PromptCCD *w*/GMP	57.10	61.14	55.70	64.10	51.00	64.70	47.67	38.67	49.47	**56.29**	**50.27**	**56.62**
Class Split: 1:2:3:4 (increasing)												
GCD [13]	52.89	63.21	51.86	53.94	53.67	53.95	45.49	33.00	45.21	50.77	49.96	50.34
Grow & Merge [19]	50.40	44.64	50.98	44.48	36.33	44.85	41.89	52.83	39.70	45.59	44.60	45.18
PromptCCD *w*/GMP	50.21	63.57	48.88	49.96	60.50	49.47	57.00	58.00	56.80	**52.39**	**60.69**	**51.72**

S8 Why Finetune the Final Block of DINO for CCD?

We analyze the number of learning parameters for each compared model and explain why the final block of our backbone is finetuned. Our motivation is to repurpose self-supervised vision foundation models for CCD. We choose DINO [2,10] as our vision foundation model to tackle CCD. DINO is a transformer-based vision foundation model pretrained on ImageNet-1K [12] with a resolution of $224 * 224$ pixels. The model is trained in a self-supervised manner (no label information) with around $86M$ parameters. Self-supervised models have been widely adopted in both NCD [4] and GCD [13] literature so far. Thus, we follow the GCD literature to use the strong DINO's self-supervised pretrained model for all compared models. We finetune the final block of its backbone and report the number of learnable parameters for each model in Tab. R. Our model's learnable parameters consist of two parts: the final block of the backbone and the parameter from GMP's GMM. The latter only accommodates $\{(2 * |\hat{z}_i| + 1) * C\}$ parameters, where C is the number of components, and $|\hat{z}_i|$ is the feature size of the `[CLS]` token, which in this case is 768. Compared with PromptCCD-B $w/\{$L2P, DP$\}$, our model's learnable parameters are only 0.33% higher when $C = 100$, which is still efficient. Moreover, in terms of the size of the prompt embedded to the backbone model, our design are much more efficient as we only embed $\{|\text{top-k}| * |\hat{z}_i|\}$ which is notably smaller compared to L2P's method, *i.e.*, $\{|\text{top-k}| * |\hat{z}_i| * L_{pp}\}$, where L_{pp} is the prompt pool's token length and DP's method, *i.e.*, $\{|\text{top-k}| * |\hat{z}_i| * L^G_{pp}\}$ for the *G-prompt* and $\{|\text{top-k}| * |\hat{z}_i| * L^E_{pp}\}$ for its *E-Prompt*, where L^G_{pp} is the prompt pool's token length for task-invariant prompt while L^E_{pp} is the token length for task-specific prompts. Here, we highlight that our proposed model only requires a minimal prompt size embedded into the backbone model. Each prompt token represents the class prototype for each category, providing strong guidance for CCD.

Table R: Information on learnable parameters for each compared model.

Method	Learnable Parameters	≈ Total Parameters
Orca [1]	$7.1M$ f_θ; $6.5M$ Classification head	$13.6M$
GCD [13]	$7.1M$ f_θ; $23.1M$ ϕ	$30.2M$
SimGCD [17]	$7.1M$ f_θ; $6.5M$ Classification head	$13.6M$
Grow & Merge [19]	$7.1M$ f_θ; $23.1M$ ϕ; $0.031M$ Cluster head	$30.2M$
PA-CGCD [7]	$7.1M$ f_θ; $0.077M$ Proxy anchor	$7.2M$
MetaGCD [18]	$7.1M$ f_θ; $23.1M$ ϕ	$30.2M$
PromptCCD-B w/L2P (Ours)	$7.1M$ f_θ; $23.1M$ ϕ; $0.046M$ L2P	$30.2M$
PromptCCD-B w/DP (Ours)	$7.1M$ f_θ; $23.1M$ ϕ; $0.045M$ DP	$30.2M$
PromptCCD w/GMP (Ours)	$7.1M$ f_θ; $23.1M$ ϕ; $\{1537 * C\}$ GMP	$30.3M$ @ $C = 100$

L2P and DualPrompt [15, 16] are prompt-based models designed for the supervised continual learning task. Both models freeze the backbone model and train the linear classifier in a supervised manner. Our CCD model $\mathcal{H}_\theta : \{\phi, f_\theta\}$ consists of ϕ, an MLP projection head, and $f_\theta : \{f_e, f_b\}$ a transformer-based feature backbone that includes an input embedding layer f_e and self-attention blocks f_b. During training, we optimize both the final block of f_b and the projection head ϕ. The projection head serves its purpose solely during contrastive learning and is omitted in the final categorization process, where only the features from f_b are utilized. Hence, freezing the backbone entirely is not feasible since it would keep the backbone unchanged even after training. To validate the necessity of finetuning the final block of the backbone, we experiment with two frozen DINO models. The first model is the default frozen DINO backbone with no prompt module. For this model, we do not perform any training strategy and directly use it to extract $\hat{z}$ features. The second model is the frozen DINO backbone coupled with a learnable L2P prompt pool. For this model, we follow the exact training procedure similar to the baseline model but keep the backbone frozen. We compare these two frozen models with both our finetuned baseline and proposed models as shown in Tab. S. By comparing the performance of the frozen models and the finetuned models, we can see that our finetuned model substantially outperforms the frozen models. Furthermore, based on the results obtained from finetuning our models on the CUB dataset, we observe that our models exhibit improved generalization compared to the DINO foundation model when applied to previously unseen datasets. This further validates the design choice of our method.

Table S: Comparison between the fully frozen models and the finetuned (final block) models.

	Method	Stage 1 *ACC* (%)			Stage 2 *ACC* (%)			Stage 3 *ACC* (%)			Average *ACC* (%)		
		All	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*	*All*	*Old*	*New*
C100	Frozen DINO [2]	64.87	71.43	60.29	55.42	66.67	53.27	49.08	66.19	46.08	56.45	68.10	53.21
	Frozen DINO *w*/L2P	65.08	73.39	59.26	55.43	64.10	53.69	49.52	67.05	46.17	56.67	68.18	53.04
	PromptCCD-B *w*/L2P (Ours)	65.93	80.20	55.94	56.72	70.76	54.04	51.55	66.67	48.90	58.07	72.54	52.96
	PromptCCD *w*/GMP (Ours)	70.69	80.90	63.54	64.08	73.14	62.35	57.73	72.67	55.12	**64.17**	75.57	60.34
IN-100	Frozen DINO [2]	68.75	71.90	66.86	70.43	73.57	69.64	62.57	74.29	59.83	67.25	73.25	65.44
	Frozen DINO *w*/L2P	76.71	77.80	75.77	64.33	67.05	63.24	63.70	76.86	61.40	68.24	73.90	66.80
	PromptCCD-B *w*/L2P (Ours)	75.80	83.84	70.17	70.95	82.48	68.75	61.19	78.67	58.13	69.31	81.66	65.68
	PromptCCD *w*/GMP (Ours)	79.56	84.24	76.29	78.58	79.71	78.36	70.33	81.33	68.40	**76.16**	81.76	74.35
Tiny	Frozen DINO [2]	55.71	65.00	52.00	45.80	56.79	43.00	46.23	55.85	42.83	49.25	59.21	45.94
	Frozen DINO *w*/L2P	62.02	66.31	59.01	52.20	61.00	50.52	46.42	54.81	44.95	53.54	60.70	51.49
	PromptCCD-B *w*/L2P (Ours)	69.46	75.24	65.41	54.64	65.43	52.58	44.88	59.43	42.33	56.33	66.70	53.44
	PromptCCD *w*/GMP (Ours)	68.67	72.84	65.76	59.69	65.67	58.55	57.16	61.10	56.47	**61.84**	66.54	60.26
CUB	Frozen DINO [2]	41.60	74.22	31.37	31.27	68.57	23.23	44.77	62.09	37.99	39.21	68.29	30.86
	Frozen DINO *w*/L2P	40.25	75.71	28.40	30.63	73.57	21.52	45.99	65.36	38.44	38.95	71.54	29.45
	PromptCCD-B *w*/L2P (Ours)	56.65	73.93	45.11	47.75	74.29	42.12	47.32	71.43	42.62	50.57	73.22	43.28
	PromptCCD *w*/GMP (Ours)	57.08	75.00	45.11	47.38	75.00	41.52	61.89	76.43	59.05	**55.45**	75.48	48.56

S9 More Qualitative Results

We further visualize the feature representation generated by our method on ImageNet-100 [12], TinyImageNet [8], and CUB [14] datasets, using t-SNE [9] to project the high-dimensional features of $\{D^l, D^u_t\}$ in each stage into a low-dimensional space. The qualitative visualization can be seen in Fig. A, where data points of the same color indicate that the instances belong to the same category. Moreover, for stage $t > 0$, we only highlight the data points belonging to unknown novel categories. It is observed that across stages and datasets, our cluster features are discriminative.

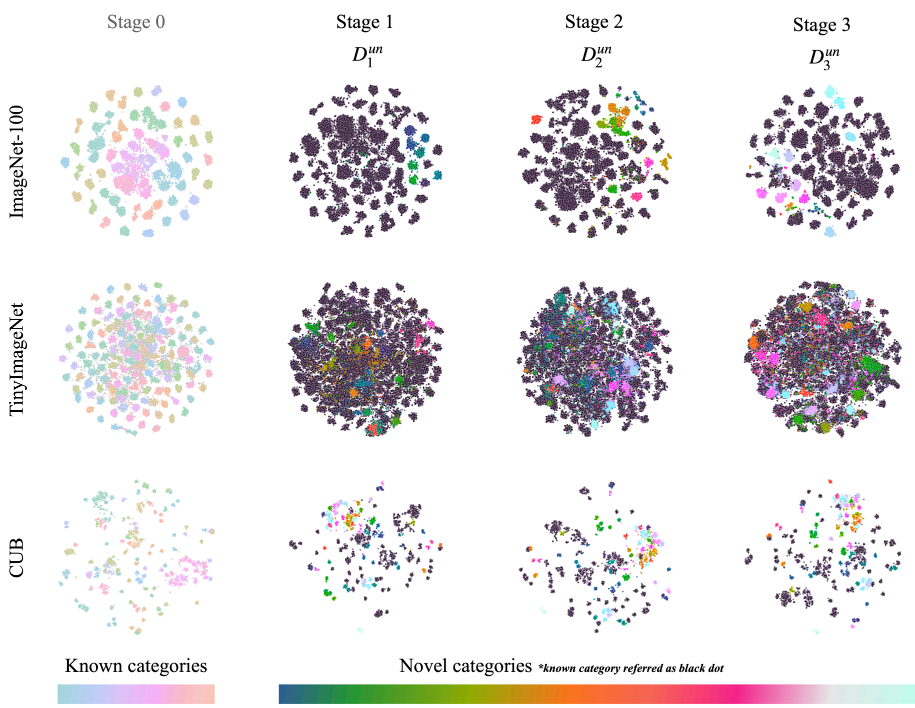

Fig. A: TSNE visualization of ImageNet-100, TinyImageNet, and CUB datasets with features from our PromptCCD w/GMP on each stage.

S10 Broader Impacts and Limitations

Category discovery technologies can significantly impact various industries and applications, such as drug discovery and materials discovery. Our proposed framework has been shown to reduce forgetting while being robust enough to discover new classes in a continual learning setting. However, there may be potential negative social impacts, such as when the model learns improper prior knowledge or the data contains unwanted bias, leading to misinformation in society. Currently, there is still no reliable mechanism to prevent such situations from happening. Therefore, having proper priors and managing data distribution is important to prevent the model from making corrupted predictions. Additionally, like other efforts on handling sequential unlabelled data, our system may accumulate errors over time as we do not have any specific regulation when dealing with longer time steps and potential categories with few samples at a given time step.

References

1. Cao, K., Brbić, M., Leskovec, J.: Open-world semi-supervised learning. In: ICLR (2022)
2. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV (2021)
3. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Automatically discovering and learning new visual categories with ranking statistics. In: ICLR (2020)
4. Han, K., Rebuffi, S.A., Ehrhardt, S., Vedaldi, A., Zisserman, A.: Autonovel: Automatically discovering and learning novel visual categories. IEEE TPAMI (2021)
5. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: CVPR (2020)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR (2016)
7. Kim, H., Suh, S., Kim, D., Jeong, D., Cho, H., Kim, J.: Proxy anchor-based unsupervised learning for continuous generalized category discovery. In: ICCV (2023)
8. Le, Y., Yang, X.: Tiny imagenet visual recognition challenge. CS 231N (2015)
9. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. JMLR (2008)
10. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. TMLR (2023)
11. Roy, S., Liu, M., Zhong, Z., Sebe, N., Ricci, E.: Class-incremental novel class discovery. In: ECCV (2022)
12. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. IJCV (2015)
13. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Generalized category discovery. In: CVPR (2022)
14. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset. California Institute of Technology (2011)
15. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: ECCV (2022)
16. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: CVPR (2022)
17. Wen, X., Zhao, B., Qi, X.: Parametric classification for generalized category discovery: A baseline study. In: ICCV (2023)
18. Wu, Y., Chi, Z., Wang, Y., Feng, S.: Metagcd: Learning to continually learn in generalized category discovery. In: ICCV (2023)
19. Zhang, X., Jiang, J., Feng, Y., Wu, Z.F., Zhao, X., Wan, H., Tang, M., Jin, R., Gao, Y.: Grow and merge: A unified framework for continuous categories discovery. In: NeurIPS (2022)
20. Zhao, B., Wen, X., Han, K.: Learning semi-supervised gaussian mixture models for generalized category discovery. In: ICCV (2023)