Appendix

In this supplementary material, we present implementation details (**Appendix A**) as well as further details on the different evaluation protocols we use (**Appendix B**). We also present additional analysis and experiments, specifically:

- **Appendix C.1** presents extended results for three different teacher combinations as well as results studying the impact of teacher dropping and the ladder of projectors independently of each other for all scenarios.
- **Appendix C.2** presents results when using UNIC models with pre-existing classifiers (plug-and-play).
- **Appendix C.3** presents results for distillation using only synthetic images from ImageNet-SD [53] dataset.
- **Appendix C.4** presents results when we distill the four teachers into a ViT-Small architecture.
- **Appendix C.5** details feature statistics on the CLS and patch tokens for the four teachers.
- **Appendices C.6 and C.7** present ablations regarding the expandable projectors and teacher dropping, respectively.
- **Appendix C.8** presents an analysis on the utilization of weights and features for the task of semantic segmentation.

A Implementation details

Data. We train all models on ImageNet-1K [52] without using image labels. For data augmentation we use random resized crop to produce 224×224 images, then apply random horizontal flip, color jitter, grayscale, Gaussian blur, and solarization, mostly following [62].

Models. All teachers and student models have the same encoder architecture: a ViT-Base [14] with patch size 16. For teachers, we download the official weights for their encoders from the authors' repositories.[4] d_h and d_h^l for ladder of projectors (LP) are set to 3072 and 768, respectively. For teacher dropping regularization (*tdrop*), we use image-level dropping with a probability of 0.25 when distilling from two teachers, and 0.5 when distilling from four teachers.

Optimization. Unless otherwise stated, models are trained for 100 epochs. When we use teacher dropping regularization and drop teacher losses, we train for longer, *i.e.* 200 epochs. It is worth noting that training the base model for double the epochs only shows small improvements.

[4] DINO: https://github.com/facebookresearch/dino
DeiT-III: https://github.com/facebookresearch/deit
iBOT: https://github.com/bytedance/ibot
dBOT-ft: https://github.com/liuxingbin/dbot

As for the distillation loss, we minimize the combination of cosine and smooth-ℓ_1 losses between the outputs of student ($\boldsymbol{s}$) and teacher ($\boldsymbol{t}$):

$$\mathcal{L}^{cos}(\boldsymbol{s},\boldsymbol{t}) = 1 - \frac{\boldsymbol{s}\cdot\boldsymbol{t}}{||\boldsymbol{s}||_2 \times ||\boldsymbol{t}||_2}, \tag{5}$$

$$\mathcal{L}^{sl1}(\boldsymbol{s},\boldsymbol{t}) = \begin{cases} 0.5 \times ||\boldsymbol{s}-\boldsymbol{t}||_2^2, & \text{for } ||\boldsymbol{s}-\boldsymbol{t}||_1 < 1, \\ ||\boldsymbol{s}-\boldsymbol{t}||_1 - 0.5, & \text{otherwise.} \end{cases} \tag{6}$$

We use the AdamW [34] optimizer, with a learning rate of 3e−4, weight decay of 3e−2, batch size of 1024 split across 4 GPUs. We apply a linear warmup for the learning rate during the first 10 epochs, then decrease it with a cosine schedule [33].

Zero-shot classification experiment. In our experiment using arbitrary models (DINOv2 and MetaCLIP) as teachers (Tab. 2), we evaluate our UNIC model's performance on zero-shot classification. Since the feature space dimensionality of UNIC is different from the features output by the MetaCLIP text encoder, we further used the projector of the MetaCLIP teacher during inference as a way of making the feature spaces compatible. This was *the only* experiment where we did not utilize the UNIC encoder features directly.

B Further details on the evaluation protocols

We perform a range of downstream tasks to evaluate the performance of models, including image classification on ImageNet-1K [52] and 15 transfer datasets, semantic segmentation on ADE-20K [80], and depth estimation on NYUd [57].

Image-level classification tasks. We measure performance on the ImageNet-1K validation set [52], on ImageNet-v2 [47], an alternative validation set for ImageNet, as well as on two datasets for measuring performance under domain shift, *i.e.* ImageNet-R [20] and ImageNet-Sketch [64].

We measure transfer learning performance on 15 datasets: 5 ImageNet-CoG levels [55] tailored for concept generalization, 8 small-scale fine-grained datasets (Aircraft [35], Cars196 [28], DTD [12], EuroSAT [19], Flowers [38], Pets [40], Food101 [6], SUN397 [69]), and two long-tail datasets (iNaturalist-2018 and 2019 [63]).

All tasks are formulated as classification tasks using linear probes attached directly to frozen encoder outputs $\boldsymbol{z}$. Each linear probe is trained separately for each dataset. We follow [54] and train linear logistic regression classifiers on top of encoder outputs. For all models (both teachers and students), we extract features from the CLS token, except for dBOT-ft, which does not include a CLS token. Following the original implementation of dBOT-ft [31], we extract the global average pooling (GAP) features instead. We then train a linear classifier using pre-extracted features, *i.e. we do not use data augmentation at this stage.* This is the reason why we report slightly lower performance on the ImageNet-1K validation set for our teacher models via this approach, *i.e.* compared to the

Table A: Distillation from different teacher combinations. We report results on four task axes for different distillation setups and teacher combinations: Distilling from a single teacher (rows 5-8), distillation from DINO & DeiT-III (rows 9-12), from iBOT & dBOT-ft (rows 13-16), and from all four teachers (rows 17-20). We report results for the strong "Base setup", *i.e.* our basic distillation setup enhanced with feature standardization and dedicated projector heads for CLS/patch tokens (row 6 of Tab. 1 from the main paper) as well as when using the proposed ladder of projectors (LP) and teacher dropping regularization (tdrop) separately on top of the base setup. Finally, we report performance using both LP and *tdrop* (UNIC models). The best performance over each column among the methods in each group is bolded.

	Method	IN-val top-1 (↑)	Transfer top-1 (↑)	Segmentation mIoU (↑)	Depth RMSE (↓)
Teachers					
1.	DINO	77.7	**72.4**	30.4	0.570
2.	iBOT	79.2	**72.4**	**36.6**	**0.524**
3.	DeiT-III	83.6	68.5	32.3	0.589
4.	dBOT-ft	**84.0**	70.7	32.8	0.616
Distillation from a single teacher					
5.	DINO	77.3	**72.9**	31.2	0.568
6.	DeiT-III	83.1	71.6	35.4	0.571
7.	iBOT	79.0	**72.9**	**36.9**	**0.531**
8.	dBOT-ft	**83.4**	72.3	35.9	0.563
Distillation from DINO & DeiT-III					
9.	Base setup	82.2	74.1	36.9	0.551
10.	+ LP	82.7	**74.2**	37.4	0.546
11.	+ *tdrop* (no LP)	83.0	74.0	36.7	0.553
12.	UNIC	**83.1**	73.9	**37.5**	**0.545**
Distillation from iBOT & dBOT-ft					
13.	Base setup	82.7	74.4	39.1	0.518
14.	+ LP	83.2	**74.8**	**39.7**	**0.505**
15.	+ *tdrop* (no LP)	83.5	74.3	38.4	0.525
16.	UNIC	**83.8**	74.5	38.9	0.515
Distillation from all four teachers					
17.	Base setup	82.8	74.5	38.5	0.539
18.	+ LP	83.3	**75.1**	**39.7**	0.518
19.	+ *tdrop* (no LP)	83.6	74.7	38.5	0.522
20.	UNIC	**83.8**	**75.1**	39.6	**0.511**

performances reported in the respective papers. For fairness, we follow this process also for all models (including teachers and students), so that linear probing setups are identical in both cases. Hyper-parameters for the linear classifiers are tuned using Optuna [2] and scikit-learn [41].

Dense prediction tasks. Semantic segmentation and depth estimation are dense prediction tasks, both formulated as classification tasks in this work, and solved following the simple setup proposed in [39]. It uses features from patch

tokens, extracted from the last output layer of the frozen encoder and used as input to a linear prediction head. For semantic segmentation, the linear head is trained to predict class logits from a patch token. This yields a 32×32 logit map, which is further upsampled via bilinear interpolation to the resolution of 512×512 to obtain a segmentation map.

For depth estimation, the features extracted from the last layer of the frozen encoder are first upsampled via bilinear interpolation by a factor of 4, then concatenated along the feature dimension with the CLS token, and finally used as input to a linear layer. Depth prediction is treated as a soft classification task using AdaBins [5] with 256 uniformly distributed bins.

Reporting a performance summary over all tasks. As metrics vary across tasks (*i.e.* top-1 accuracy for classification, mIoU for segmentation and RMSE for depth estimation), in Fig. 1 of the main paper we report *relative* performance for each task, which is calculated on each task as the difference between the performance of our UNIC model distilled from four teachers to that of the best teacher, divided by that same best performance.

C Extended analysis and results

C.1 Extended results and component ablations

In Tab. A we report results when distilling from two sets of teachers, as well as distilling from all four. We report results for a number of distillation configurations: a) a "base setup", which is our basic distillation setup detailed in Section 3.1 of the main paper, plus feature standardization and dedicated projectors for CLS and patch tokens; a very strong baseline to beat, b) using a ladder of projectors (LP) over the base setup, c) using teacher dropping (*tdrop*) over the base setup and d) results for UNIC models, *i.e.* models trained using the base setup plus a ladder of projectors and teacher dropping regularization.

We see that both LP and *tdrop* show improved gains, with LP maximizing the gains for dense prediction tasks, but still lacking on ImageNet-1K, the task most complementary to the rest for the selected teachers. When using *tdrop* without LP, we see that it can achieve strong balance over the tasks that the teachers are complementary at, but dense prediction performance is not really improved. When using both modifications together, we see that we get the best possible results overall, with ImageNet-1K performance now reaching the performance of the best teacher.

Distilling from a single teacher. In Tab. A we also show results after using our distillation setup to distil from each teacher independently. By simply using a form of *self-distillation* we see that the transfer learning performance of DeiT-III and dBOT-ft, the two models tuned for ImageNet-1K, increases significantly. One explanation is that since the features at the output of the student encoder are followed by a projector, they might have become more generic than the ones from teachers, which are tailored for the task. We see similar but smaller gains on that axis also for the self-supervised models DINO and iBOT.

Table B: Plug-and-play performance on the ImageNet-1K validation set. For our UNIC models distilled from either one of the teacher pairs or all four of them, we report their logistic regression (LogReg) and plug-and-play evaluations using the pre-existing classifiers from the best supervised teacher (DeiT-III for the first row which reaches 83.5 top-1 accuracy, dBOT-ft for the second and third rows, which reaches 84.5 top-1 accuracy). For LogReg (which is our default evaluation protocol for image classification tasks in this paper), we train linear logistic regression classifiers on top of pre-extracted encoder representations. For plug-and-play, we use the pre-existing ImageNet-1K classifiers from the teacher which are fed from the projected student features; this does not require any task-specific training for the student.

Model	LogReg	Plug-and-play
UNIC (DINO & DeiT-III)	83.1	83.3
UNIC (iBOT & dBOT-ft)	83.6	83.8
UNIC (4 teachers)	83.8	84.0

C.2 Results with pre-existing classifiers (plug-and-play)

The student is trained together with teacher-specific projector(s) that mimic the teacher features. It is thus possible to directly use a task head, learned with teacher features, and directly plug it on top of the corresponding teacher projectors we learn together with the student encoder. Tab. B shows the results on the ImageNet-1K validation set when using the pre-existing classifiers from the public DeiT-III and dBOT-ft models as well as using linear probes trained with our protocol.

We see that the *plug-and-play* scenario can lead to higher accuracies using the projectors rather than the original student features. This shows that heads trained for a specific teacher can be directly used without any retraining. The higher accuracies can also be explained by the fact that our evaluation protocol does not include data augmentation for efficiency reasons (see Appendix B). Also note that the projectors add extra parameters on top of the backbone.

C.3 Distilling using synthetic images from ImageNet-SD

In a recent study, Sariyildiz *et al*. [53] replace the ImageNet-1K dataset for supervised training with *ImageNet-SD*, an ImageNet clone composed of Stable Diffusion [48] images obtained using the ImageNet class names as prompts.

In Tab. C we report results when using this dataset for distillation instead of ImageNet-1K. We see that the UNIC model distilled exclusively on synthetic images is outperforming the best teacher on transfer learning and semantic segmentation. Similar to the observations in [53], we also see that performance on classifying the dataset classes decreases. The decrease is however relatively small: the student is better than teachers like iBot or DINO, and outperformed only by the teacher optimized for this specific classification task.

Table C: Multi-teacher distillation using synthetic data. We replace ImageNet-1K with ImageNet-SD [53] for distilling UNIC models. ImageNet-SD is an ImageNet-sized dataset composed of synthetic images generated with Stable Diffusion [48] using the ImageNet class prompts; we refer the reader to [53] for more details.

	Method	IN-val top-1 (↑)	Transfer top-1 (↑)	Segmentation mIoU (↑)	Depth RMSE (↓)
Teachers (Trained on ImageNet-1K)					
21.	DINO	77.7	72.4	30.4	0.570
22.	iBOT	79.2	72.4	36.6	0.524
23.	DeiT-III	83.6	68.5	32.3	0.589
24.	dBOT-ft	**84.0**	70.7	32.8	0.616
*Multi-teacher distillation using **ImageNet-1K** or **ImageNet-1K-SD***					
25.	**UNIC**	**83.8**	**75.1**	**39.6**	**0.511**
26.	**UNIC-SD**	81.7	**74.7**	**37.8**	**0.528**

Table D: Distilling four ViT-Base/16 teachers into different student architectures. The "Num. Params." column refers to the number of trainable parameters in the encoder of the student architecture.

Method	Student Architecture	Num. Params.	IN-val top-1 (↑)	Transfer top-1 (↑)	Segmentation mIoU (↑)	Depth RMSE (↓)
UNIC	ViT-Base/16	85.8M	83.8	75.1	39.6	0.511
UNIC	ViT-Small/16	21.7M	81.4	71.6	36.1	0.564

C.4 Distilling into a ViT-Small student

In Tab. D we report results when distilling the four teachers into a smaller student architecture, ViT-Small/16. Our ViT-Small UNIC model also matches the performance of a ViT-Small DeiT-III on ImageNet 1K.[5]

C.5 Statistics for CLS and patch tokens

In Tab. E we report norm and standard deviation for CLS and patch token features from all our teacher models, computed on the ImageNet-1K validation set. We see large variations in the moments, not only across teachers but also across CLS and patch tokens of the same model.

C.6 Expendable projector ablations

Top-only projector heads. We employ such projector heads when not using the ladder of projectors. In Tab. F, we vary the number of hidden layers in top-only projector heads when distilling from DINO and DeiT-III, and check how

[5] See https://github.com/facebookresearch/deit/blob/main/README_revenge.md

Table E: Feature statistics obtained on the the ImageNet-1K validation set. For each teacher, we extract their encoder outputs, as we do in our evaluations. "CLS" refers to features of the CLS token, while "Patch" refers to patch token features, where the statistics are computed after global average pooling (GAP) applied spatially. "Avg. norm per sample" (resp. "Avg. std per sample") is the average ℓ_2 norm (resp. standard deviation) of features computed over samples. "Avg. std per dimension" is the average standard deviation computed over dimensions. dBOT-ft does not contain a CLS token. When we distill from dBOT-ft, we use its GAP features.

Model	Feature Type	Avg. norm per sample	Avg. std per sample	Avg. std per dimension
DINO	CLS	66.6	2.4	2.2
DeiT-III	CLS	23.3	0.8	0.5
iBOT	CLS	69.9	2.5	2.3
DINO	Patch	31.3	1.1	0.5
DeiT-III	Patch	26.2	0.9	0.5
iBOT	Patch	36.3	1.3	0.9
dBOT-ft	Patch	9.8	0.4	0.4

they impact performance across all tasks. Hidden (d_h) and output layer dimensions are set to 3072 and 768, similar to the original ViT-Base specification [14]. We see that having 1 hidden and output layers (which is highlighted in gray) is the best for ImageNet-1K classification and NYUd depth estimation.

Ladder of projectors . When using the ladder of projectors, features from intermediate blocks of the student encoder are projected with a teacher-specific MLP and summed together with the outputs of the projector attached to the last encoder layer. In Tab. G, we ablate the number of hidden dimensions d_h^l in the MLPs of intermediate blocks, as well as which intermediate blocks are considered. Regarding the hidden dimensions, we see that performance improves for ImageNet-1K as the hidden dimension increases, up to a plateau after 384 for semantic segmentation. To keep the number of parameters relatively small, we thus chose 768. Regarding which blocks to consider, the impact is overall limited as long as sufficient blocks are considered, and considering all of them lead to the best performance on ImageNet-1K.

C.7 Teacher dropping ablations

Impact of *tdrop* granularity and probability. In Tab. H, we study the impact of the teacher dropping probability p on performance, when *tdrop* is used with and without LP and varying the dropping probability between 0 and 1. We see that increasing the dropping probability (*i.e.* training with sparser teachers) leads to generally better performance on ImageNet-1K, while, lower probability leads to better performance on the remaining of the tasks (for transfer learning, semantic segmentation and depth estimation). Specifically, higher dropping probability p improves performance on the tasks where the "underlearned" teacher

Table F: Architecture of the student projector used in the absence of the ladder of projectors. Results are reported for distillation from DINO and DeiT-III without using *tdrop* but using feature standardization and dedicated projectors. We vary the number of hidden and output layers in the projectors. Number of units for hidden and output layers are 3072 and 768, respectively. The row corresponding to the default setup in our experiments is colored in light gray.

Projector		IN-val	Transfer	Segmentation	Depth
Hidden L.	Output L.	top-1 (↑)	top-1 (↑)	mIoU (↑)	RMSE (↓)
–	–	81.1	73.0	34.1	0.564
–	1	81.5	73.1	35.3	0.567
1	1	82.2	74.1	36.9	0.551
2	1	81.8	74.2	36.9	0.559
3	1	81.1	74.2	37.0	0.559

Table G: Architecture for the ladder of projector. We vary the hidden dimension of the non-final block (768 by default) as well as which intermediate blocks are connected in the ladder (by default, all, *i.e.* $\{1, .., 11\}$). Results are reported for distillation from DINO and DeiT-III without using *tdrop* but using feature standardization and dedicated projectors. The row corresponding to the default setup in our experiments is colored in light gray.

Hidden dim.	Blocks	IN-val top-1 (↑)	Transfer top-1 (↑)	Segmentation mIoU (↑)	Depth RMSE (↓)
64	$\{1, .., 11\}$	81.9	74.5	36.1	0.549
192	$\{1, .., 11\}$	82.3	74.5	36.9	0.540
384	$\{1, .., 11\}$	82.5	74.4	37.8	0.547
768	$\{1, .., 11\}$	82.7	74.2	37.4	0.546
1536	$\{1, .., 11\}$	82.7	74.5	37.7	0.544
768	$\{6\}$	82.0	74.6	36.7	0.545
768	$\{3, 6, 9\}$	82.3	74.3	37.3	0.545
768	$\{9, 10, 11\}$	82.0	74.4	37.1	0.542
768	$\{2, 4, 6, 8, 10\}$	82.5	74.4	37.8	0.541

excells, *i.e.* DeiT-III and ImageNet for the case of DINO and DeiT-III teachers. One can therefore adjust p according to the desired performance on the tasks of the teacher(s) with generally higher loss.

In the same table, we further study the impact that *tdrop* granularity has, *i.e.* when dropping losses on the image or patch level, with the former being the default in all our experiments. We see no noticeable gains when dropping teachers at the patch level.

Comparing teacher dropping regularization to alternatives. In Tab. I, we compare *tdrop* to AdaLoss [24], another automatic loss balancing technique, and manual balancing of losses when distilling from all four teachers. For manual balancing, it is computationally demanding to find the optimal teacher weights due to its combinatorial nature. We choose 5 different intuitive combinations

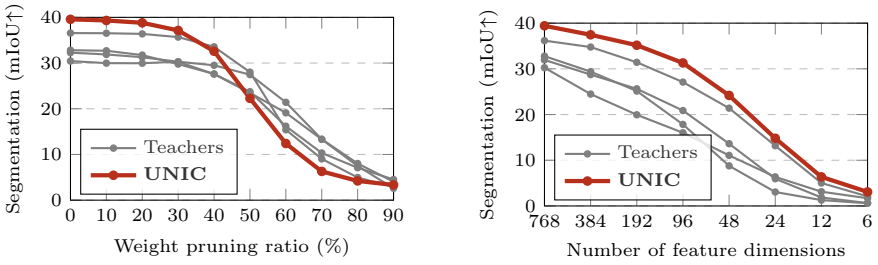
Table H: Impact of *tdrop* probability and granularity. We vary the probability between 0 and 1, and the granularity to be either at the image or patch level. We show results for distillation from iBOT & dBOT-ft, without using a ladder of projectors. We use feature standardization and dedicated projectors in all cases.

tdrop gran.	*tdrop* prob.	LP	IN-val top-1 (↑)	Transfer top-1 (↑)	Segmentation mIoU (↑)	Depth RMSE (↓)
Image	0.00	–	83.0	74.4	39.1	0.518
Image	0.25	–	83.1	74.3	38.7	0.522
Image	0.50	–	83.5	74.3	38.4	0.525
Image	1.00	–	83.5	73.9	37.9	0.530
Patch	0.50	–	83.2	74.3	38.7	0.532
Patch	1.00	–	83.3	74.1	38.0	0.533
Image	0.00	✓	83.2	74.8	39.7	0.505
Image	0.25	✓	83.6	74.5	39.4	0.506
Image	0.50	✓	83.8	74.5	38.9	0.515
Image	1.00	✓	83.7	73.6	38.1	0.530

to see the relative impact of each teacher. We see that *tdrop* achieves significantly better performance than AdaLoss on ImageNet-1K and segmentation, while being comparable to AdaLoss on the remaining tasks. In the case of manual balancing, no single combination leads to best performance on all tasks.

Table I: Loss balancing techniques for distillation from all four teachers (DINO, DeiT-III, iBOT and dBOT-ft). We use feature standardization and dedicated projectors in all cases. The best (resp. second best) performance over each column among the methods in each group is bolded (resp. underlined). All experiments performed over the base setup, *i.e.* using feature standardization and dedicated projectors for CLS/patch tokens and without using a ladder of projector heads.

Method	IN-val top-1 (↑)	Transfer top-1 (↑)	Segmentation mIoU (↑)	Depth RMSE (↓)
Manual balancing				
DINO×**1** + DeiT-III×**1** + iBOT×**1** + dBOT-ft×**1**	82.2	**74.5**	<u>38.5</u>	0.539
DINO×**4** + DeiT-III×**1** + iBOT×**1** + dBOT-ft×**1**	80.6	74.0	36.1	0.549
DINO×**1** + DeiT-III×**4** + iBOT×**1** + dBOT-ft×**1**	83.2	74.0	37.4	0.548
DINO×**1** + DeiT-III×**1** + iBOT×**4** + dBOT-ft×**1**	81.1	74.1	38.2	<u>0.533</u>
DINO×**1** + DeiT-III×**1** + iBOT×**1** + dBOT-ft×**4**	**83.5**	74.2	38.4	**0.532**
Automatic balancing				
AdaLoss	81.9	**74.5**	38.4	0.536
Teacher dropping (*tdrop*)	<u>83.1</u>	<u>74.4</u>	**38.8**	<u>0.533</u>



(a) Pruning analysis **(b)** PCA analysis

Fig. A: Network utility analysis for semantic segmentation linear probing for the four teachers and our student UNIC distilled from all of them. For each model, before training linear probes, we either **(a)** prune their weights or **(b)** reduce the dimension of their features via PCA. We report the mIoU scores on ADE-20K. UNIC's encoder weights work together more cohesively **(a)**, and its feature space is more robust to dimensionality reduction **(b)**.

C.8 Extended results on weight and feature space utilization

In Section 4.2 of the main paper, we study the network utility for teachers and our best UNIC model in terms of the utility of their weights and CLS features for ImageNet-1K classification. We extend this analysis for semantic segmentation, this time, using patch tokens. From the results shown in Fig. A, we see that our observations from the main paper are consistent. When varying the weight pruning ratio (Fig. Aa), UNIC's performance drops significantly faster than the ones from the teachers, meaning that the weights are better utilized. When applying PCA to reduce dimension of the features (Fig. Ab), we see that the UNIC performance remains higher than the ones from the teachers, showing that it better utilizes the feature space.