Appendix

The appendix is organized as follows:

- In Sec. A1, we provide additional implementation details of our method.
- In Sec. A2, we provide additional details of our monotone scaling group and its parameterization.
- In Sec. A3, we provide the detailed proofs for our lemmas in Sec. A4, we discuss the runtime of the DEC module.
- In Sec. A5, we provide additional analysis and results.

A1. Implementation details

DEC module. For all experiments, we model the DEC module as 2 layered CNN mapping with 64 and 128 channels. We perform adaptive pooling at the end to get the desired number of monotone scaling parameters.

Vanilla Canonicalization. To train vanilla canonicalization, we discretized the monotone scale parameter into 64 different configurations for the locally scaled MNIST and object segmentation and 25 different configurations for locally scaled ImageNet. The learnable energy functions are implemented via a 3 layer CNN.

Locally Scaled Object Segmentation. The pretrained models ViT [18], Swin [38], and DINOv2 [45] are finetuned on the training set for 120 epochs. We set the initial learning rate at 1e$-$4 and scaled it by a factor of 0.7 for each 30 epoch. We use DPT [50] style segmentation head. We use the per-pixel cross-entropy loss to train each model. We set the weight for the "background" pixel class to 0.1, and the weights of all other object classes are set to 1. All baselines are finetuned for 60 epochs with an initial learning rate of 2e$-$5

Locally Scaled MNIST. Each architecture is trained for 50 epochs. We set the initial learning rate at 1e$-$4 for ResNet [25], ViT [18], DeiT [61], and BEiT [7]; 2e$-$5 for DINOv2 [45]; and 8e$-$4 for Swin [38]. All baselines are fine-tuned for 40 epochs.

Locally Scaled ImageNet. All baselines are fine-tuned following standard data augmentation practice for ImageNet finetuning [26, 38]. We use a batch size of 80 and an initial learning rate of 1e$-$7 with 5 warm-up epochs. We list common training hyperparameters in Tab. A1.

A2. Monotone scaling group

A2.1. Group axioms

Our monotone scaling set L must meet the following axioms to be considered as a group. Here, "$\cdot$" represents the group product described in Eq. (9). The 4 axioms are as follows:

- **Closure:** $\forall a, b \in L, a \cdot b \in L$
- **Associativity:** $\forall a, b, c \in L, a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- **Existence of Identity:** $\exists e \in L : \forall a \in L, \quad e \cdot a = a$
- **Existence of Inverse:** $\forall a \in L, \exists a^{-1} \in L : a \cdot a^{-1} = e.$

Hyperparameter	**Value**
Scheduler	Cosine
Weight decay	0.05
Warmup epochs	5
Mixup Alpha	0.8
Label smoothing	0.1
Random Erase Prob	0.25
Layer Decay Factor	0.75
Epochs	20

Table A1. ImageNet Finetuning Parameters

A2.2. 2D Monotone scaling group

Construction of monotone scaling group in 2D. To form the monotone scaling group on $2D$, the set

$$L_{2\text{D}} : \{l : [0,1]^2 \rightarrow [0,1]^2\}$$

should satisfy the two following properties:

- For all $x \in [0,1]^2$ there is a neighborhood $\mathcal{W} \subset [0,1]^2$ such that any function $l \in L_{2\text{D}}$ can be approximated by a linear function with Jacobian $J_l(x) \in \text{SPD}(2)$, *i.e.*, 2×2 symmetric positive definite Jacobian. This condition imposes local monotonicity on l.
- For all $l_1, l_2 \in L_{2\text{D}}$, their local Jacobian $J_{l_1}(x_1)$ and $J_{l_2}(x_2)$ commutes for $x_1, x_2 \in [0,1]^2$.

Formally, we state this in Lemma 2.

> **Lemma 2.** *The set of all locally monotone increasing functions L_{2D} with commutative Jacobian is a valid group under the binary operation of function composition.*

Proof. We provide the detailed proof in Sec. A3.2. □

Parametrization of l. We parameterize l through a set of independent monotone functions along the x and y axes via bilinear interpolation. Specifically, we decompose the function l into two functions as

$$l(x,y) = (l_X(x,y), l_Y(x,y)), \tag{A26}$$

where $l_X, l_Y : [0,1]^2 \rightarrow [0,1]$. We parameterize each of them as piecewise linear functions. To achieve this we discretize the domain $[0,1]^2$ into uniform grid $\mathcal{G} = \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} = \{x_0 = 0, x_1, \ldots, x_N = 1\}$ and $\mathcal{Y} = \{y_0 = 0, y_1, \ldots, y_M = 1\}$. We assume the set is ordered, *i.e.*, $x_i < x_j$ when $i < j$.

We define independent monotone functions along each row y_j and each column x_i of the grid $\mathcal{G}$ as follows:

For $x \in [x_{n-1}, x_n)$, the monotone function along row y_j is given by:

$$l^{y_j}(x) = \phi^{y_j}_{x_{n-1}} + \frac{\phi^{y_j}_{x_n} - \phi^{y_j}_{x_{n-1}}}{x_n - x_{n-1}} \times (x - x_{n-1}). \tag{A27}$$

Similarly, for $y \in [y_{m-1}, y_m)$, the monotone function along column x_i is given by:

$$l^{x_i}(y) = \phi^{x_i}_{y_{m-1}} + \frac{\phi^{x_i}_{y_m} - \phi^{x_i}_{y_{m-1}}}{y_m - y_{m-1}} \times (y - y_{m-1}) \tag{A28}$$

Here, ϕs are the learnable parameters and to preserve monotonicity we impose restriction $\phi^{y_j}_{x_{n-1}} \leq \phi^{y_j}_{x_n}$ $\forall j, n$ and $\phi^{x_i}_{y_{m-1}} \leq \phi^{x_i}_{y_m}$ $\forall i, m$.

Finally, we obtain $l_X(x, y)$ from $l^{y_i}s$ via linear interpolation as

$$l_X(x, y) = l^{y_{j-1}}(x) + \frac{l^{y_j}(x) - l^{y_{j-1}}(x)}{y - y_{j-1}} \tag{A29}$$

when $y \in [y_{j-1} \leq y_j)$. We defined $l_Y(x, y)$ similarly. We approximate the inverse function l^{-1} by computing the inverses of each l^{x_i} and l^{y_j} individually.

A3. Complete Proofs of Lemmas and Claims

A3.1. Proof of Lemma 1

Lemma 1. *The set of all continuous strictly monotonic increasing functions L is a group under the binary operation of function composition.*

Proof. To prove that the set of all continuous strictly monotonic increasing functions L forms a group under function composition, we need to verify four properties: closure, associativity, identity element, and inverse element.

Closure. Let $l_1, l_2 \in L$. Since l_1 and l_2 are continuous and strictly increasing, for any $x_1, x_2 \in [0, 1]$ if $x_1 < x_2$, then $l_1(x_1) < l_1(x_2)$. Thus, $l_2(L_1(x_1)) < l_2(l_1(x_2))$, showing that $l_1 \circ l_2$ is strictly increasing. Moreover, $l_1 \circ l_2$ is continuous because both are continuous.

Associativity. Function composition is inherently associative, thus satisfying the property.

Identity Element. The identity function is continuous and strictly increasing, so also an element of L

Inverse Element. Strictly monotone functions have an inverse, and the inverse is also monotonic. Thus, the inverse is also an element of L.

Therefore, we conclude that L is a group. □

A3.2. Proof of Lemma 2

Lemma 2. *The set of all locally monotone increasing functions L_{2D} with commutative Jacobian is a valid group under the binary operation of function composition.*

Proof. To verify that $L_{2\text{D}}$ forms a group, we check the following properties:

Closure: For any $l_1, l_2 \in L_{2\text{D}}$, their composition $l_1 \circ l_2$ is also a locally monotone-invertible function.

Because the local Jacobian of the composition $l_1 \circ l_2$ is

$$J_{l_1 \circ l_2}(x) = J_{l_1}(l_2(x)) \cdot J_{l_2}(x) \quad \forall x \in [0, 1]^2. \tag{A30}$$

Since $J_{l_1}, J_{l_2} \in \text{SPD}(2)$ and commutes, their product $J_{l_1} \cdot J_{l_2}$ is also a SPD matrix. The composition of invertible functions is also invertible. And the $J_{l_1 \circ l_2}$ commutes due to associativity of matrix product. Thus $l_1 \circ l_2 \in L_{2\text{D}}$.

Associativity: Function composition is inherently associative, *i.e.*, for all $l_1, l_2, l_3 \in L_{2\text{D}}$, we have

$$(l_1 \circ l_2) \circ l_3 = l_1 \circ (l_2 \circ l_3). \tag{A31}$$

Existence of Identity: The identity function l has 2×2 identity matrix as local Jacobin. Thus, it maintains all the conditions of $L_{2\text{D}}$.

Existence of Inverse: The inverse function of any $l \in L_{2\text{D}}$ can be obtained by inverting the local Jacobians. Specifically, for any $l \in L_{2\text{D}}$, the inverse l^{-1} exists and satisfies:

$$J_{l^{-1}}(\mathbf{x}) = J_l(l^{-1}(\mathbf{x}))^{-1} \in \text{SPD}(2) \quad \forall x \in [0, 1]^2 \tag{A32}$$

Furthermore, $J_{l^{-1}} J_{l_k} = J_{l_k} J_{l^{-1}}$ for any $l_k \in L_{2\text{D}}$ as

$$J_l J_{l_k} = J_{l_k} J_l \tag{A33}$$

$$\Rightarrow J_l^{-1}(J_l J_{l_k}) = J_l^{-1}(J_{l_k} J_l), \text{(left multiply by } J_l^{-1}) \tag{A34}$$

$$\Rightarrow J_{l_k} = J_l^{-1} J_{l_k} J_l \tag{A35}$$

$$\Rightarrow J_{l_k} J_l^{-1} = J_l^{-1} J_{l_k} \text{(right multiply by } J_l^{-1}). \tag{A36}$$

Thus, $l^{-1} \in L_{2\text{D}}$.

Therefore, $L_{2\text{D}}$ satisfies all group axioms, completing the proof. □

A4. Runtime of the DEC Module

We use Anderson Acceleration to approximate the fixed point of the DEC. This requires a fixed number of forward passes through the lightweight DEC module. The computational complexity of this iterative process is $\boldsymbol{O}(jT_{\text{DEC}})$, where j is a fixed number of required iterations and T_{DEC} is computation cost associated with a single forward pass of the DEC module. The hyperparameter j governs the trade-off between computational cost and the accuracy of the fixed-point approximation.

Empirically, for DINO-v2, the DEC module requires 24% (0.16 sec) of the total required time (0.66 sec) to process a batch of 128 images of size 224 × 224.

A5. Additional Results

A5.1. Additional Baselines

To evaluate the effectiveness of the DEC module in models with handcrafted hierarchical feature processing or image pyramid structures, we adapt our approach to HRViT [22] and ResFormer [60] and report the results in Tab. A2.

Method	HRVit [22]	Resformer [60]
Aug	93.22	91.04
Canon	94.93	95.27
InvL	95.91	94.92
Ours	**96.67**	**96.91**

Table A2. Hierarchical baselines on scale-MNIST

		# layers in DEC Mod.			
		1	2	3	4
# DEC Mod	1	93.59	94.47	94.04	93.22
	2	94.98	96.04	95.16	95.25
	3	96.66	96.17	96.67	96.38
	4	96.58	96.78	96.65	96.50

Table A3. Ablation on the number of DEC modules and layers per module on scale-MNIST

		Multiples (×) of grid			
		1.0	1.5	2.0	2.5
layers	2	96.61	96.95	97.06	97.94
	3	96.62	96.91	97.08	97.08

Table A4. Acc. at multiple of initial grid size of 4 with varying number of layers in DEC on scale-MNIST.

A5.2. Additional Ablation Study

We perform additional ablation studies on the scale-MNIST dataset to evaluate the effect of (i) the number of DEC modules and (ii) the number of layers within each DEC module. The results are summarized in Tab. A3. We observe that increasing the number of DEC modules, *i.e.*, repeatedly canonicalizing features throughout the network, improves performance compared to applying canonicalization only at the input level.

We provide an additional ablation study on the choice of grid size for local scaling and report the results in Tab. A4. We observe that increasing the grid size improves the performance as it allows more flexible spatial parameterization of the local scaling operations.

To assess potential side effects of scale equivariance, we report the accuracies of the adapted models on images of scale 1, *i.e.*, unmodified images in Tab. A5. We do not observe any drop in the performance.

A5.3. Additional Visualizations

Following the settings of Fig. 8, we report the results for more input images in Fig. A1. We observe that Ours is consistently better and more robust on all scales in comparison to *Base*; especially on the more extreme local scale factors.

We present visualizations of learned monotone scaling by the DEC trained on MNIST in Fig. A2. We observe that the DEC module has learned to stretch/squeeze regions of the digits. However, the exact reasoning on why such scaling is beneficial to the deep-net remains challenging. The interpretability of the choice of learned canonical elements in a group is largely underexplored in the literature.

Methods	ViT	DeiT	Swin	BEiT
Base	81.29	70.67	79.55	85.79
Aug	81.29	70.70	79.56	85.66
Canon	79.23	66.92	76.04	84.29
InvL	81.29	70.71	79.58	85.66
Ours	**81.43**	**70.92**	**79.86**	**86.04**

Table A5. Acc. on unmodified (scale-1) ImageNet images.

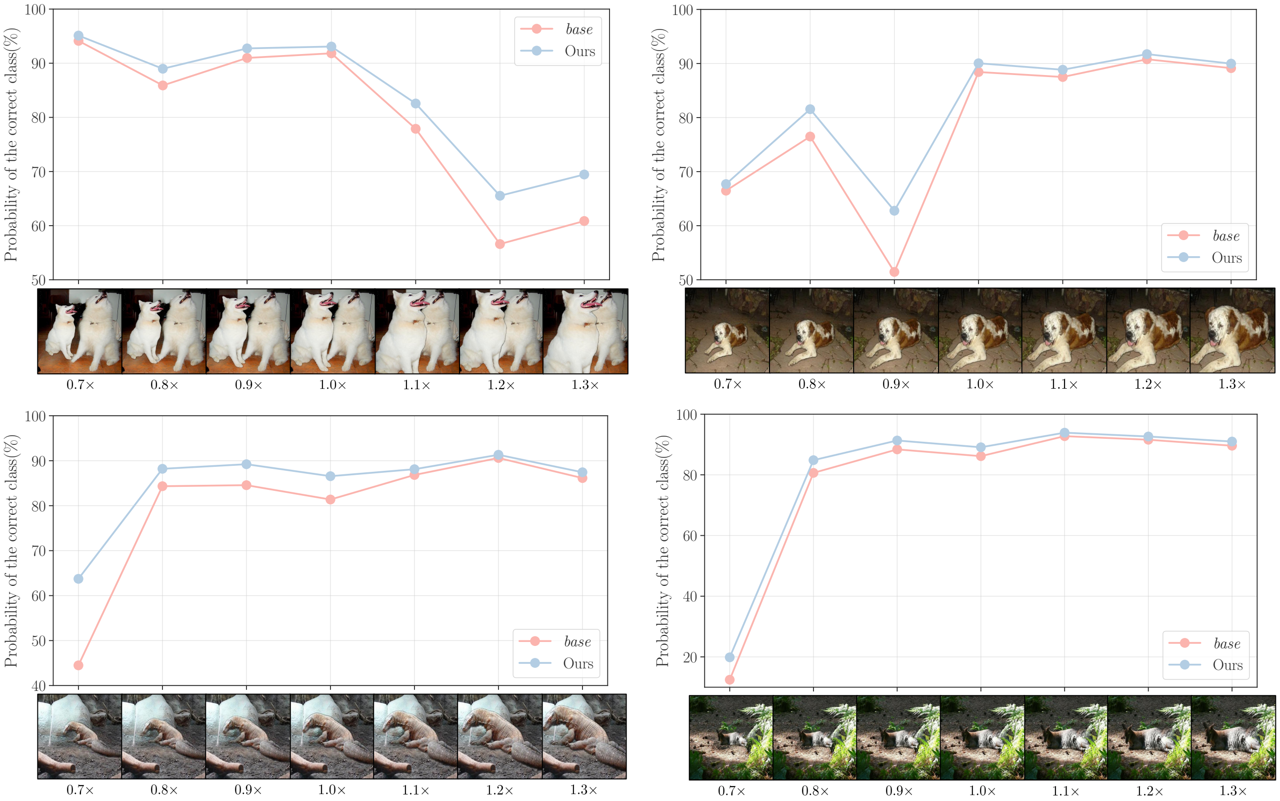

Figure A1. **Comparison on per-scale probability of correctness.** We locally scale the same input image within the range of $[0.7, 1.3]$ and report the probability of the correct class.

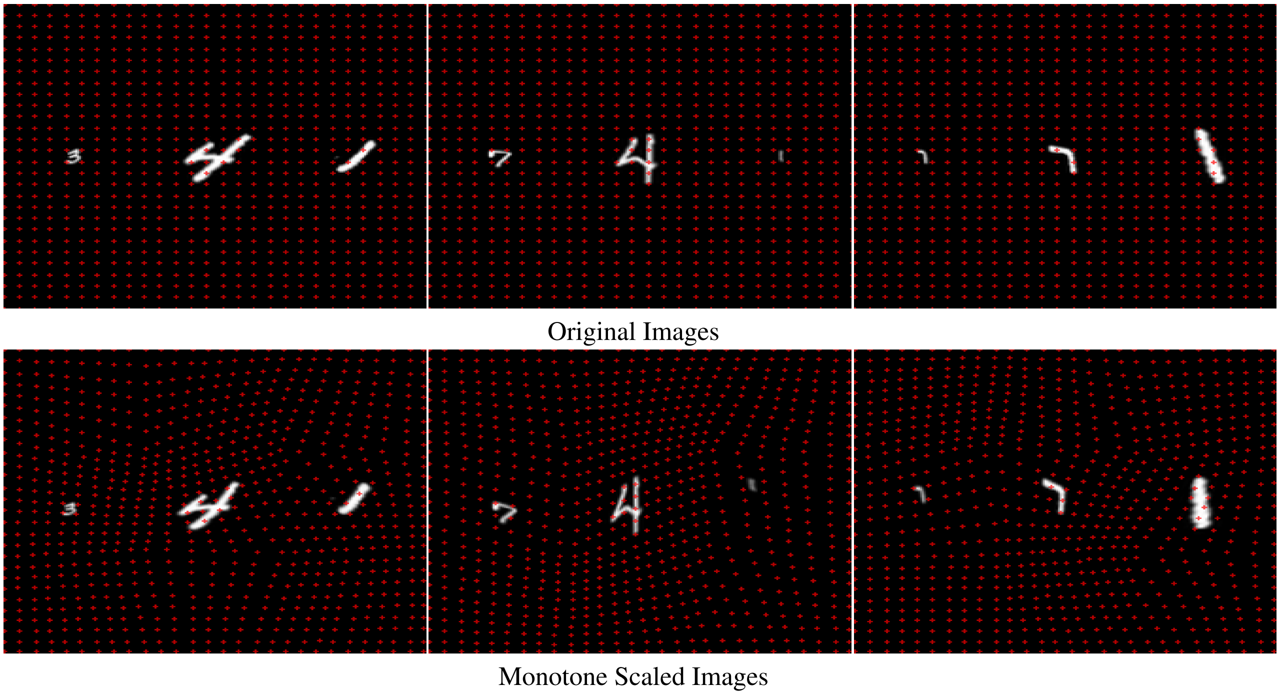

Figure A2. **Learned monotone scaling on locally scaled MNIST.** We observe that stretching/squeezing is performed on the area with digits.