LCM: Log Conformal Maps for Robust Representation Learning to Mitigate Perspective Distortion

Meenakshi Subhash Chippa[1,*], Prakash Chandra Chhipa[1], Kanjar De[2], Marcus Liwicki[1], and Rajkumar Saini[1]

[1] Luleå Tekniska Universitet, Sweden
{prakash.chandra.chhipa, rajkumar.saini, marcus.liwicki}@ltu.se
*meechi-2@student.ltu.se
[2] Fraunhofer Heinrich-Hertz-Institut, Berlin, Germany
kanjar.de@hhi.fraunhofer.de

Abstract. Perspective distortion (PD) leads to substantial alterations in the shape, size, orientation, angles, and spatial relationships of visual elements in images. Accurately determining camera intrinsic and extrinsic parameters is challenging, making it hard to synthesize perspective distortion effectively. The current distortion correction methods involve removing distortion and learning vision tasks, thus making it a multi-step process, often compromising performance. Recent work leverages the Möbius transform for mitigating perspective distortions (MPD) to synthesize perspective distortions without estimating camera parameters. Möbius transform requires tuning multiple interdependent and interrelated parameters and involving complex arithmetic operations, leading to substantial computational complexity. To address these challenges, we propose Log Conformal Maps (LCM), a method leveraging the logarithmic function to approximate perspective distortions with fewer parameters and reduced computational complexity. We provide a detailed foundation complemented with experiments to demonstrate that LCM with fewer parameters approximates the MPD. We show that LCM integrates well with supervised and self-supervised representation learning, outperform standard models, and matches the state-of-the-art performance in mitigating perspective distortion over multiple benchmarks, namely Imagenet-PD, Imagenet-E, and Imagenet-X. Further LCM demonstrate seamless integration with person re-identification and improved the performance. Source code is made publicly available at https://github.com/meenakshi23/Log-Conformal-Maps.

Keywords: Perspective Distortion · Robust Representation Learning · Self-supervised Learning

1 Introduction

Perspective distortion (PD) is a common issue in real-world imagery, complicating the development of computer vision applications. It arises from factors

like camera position, depth, focal length, lens aberrations, and rotation, which affect the projection of 3D scenes onto 2D surfaces [20]. Accurately estimating these parameters for PD correction is challenging, posing a significant barrier to robust computer vision (CV) methods. Earlier studies focused on distortion

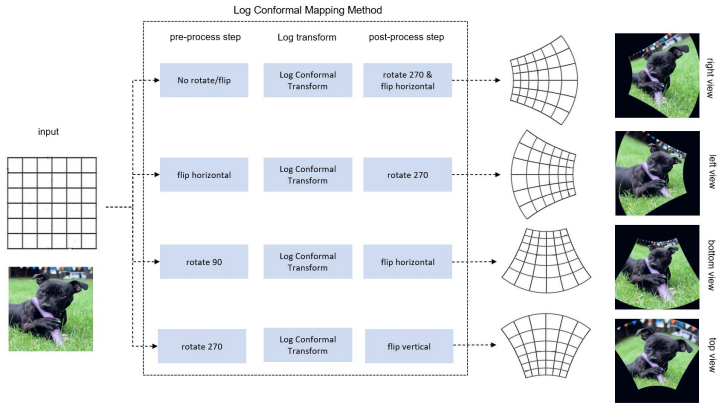


Fig. 1: Log conformal Mapping Method (LCM). LCM obtains four perspective distorted views using auxiliary operations with Log Conformal transform.

correction [18], [27], [21], which makes vision task learning multi-steps. Recent work MPD [5] provides insights on mitigating perspective distortion by synthetically mimicking the PD in the learning process using nonlinear and conformal Möbius [1] transform.

Previous research on image registration [22] demonstrates the potential of conformal log-polar mapping. Another study highlights the utility of complex logarithmic views [2] for exploring small details within extensive visual geometry contexts. Focusing on non-linear and conformal properties of log transform, we proposed Log Conformal Mapping (LCM) to mitigate perspective distortion and match state-of-the-art robustness performance. LCM method in Figure 1 demonstrates the synthesis of four perspective distorted views (left, right, top, bottom) mentioned in MPD [5]. The main contributions of this paper are:

1. To the best of our knowledge, this work is the first to employ Log Conformal Transform for synthesizing perspective distortion, introducing the Log Conformal Maps (LCM) method.
2. We present analytical and empirical evidence demonstrating that LCM, with fewer parameters, effectively mitigates perspective distortion, comparable to the Möbius Transforms-based MPD method.
2. We validate our approach through extensive experiments on multiple perspective distortion-affected benchmarks, showing improvements in the robustness of both supervised and self-supervised models and showing effectiveness on real-world application, person re-identification.

2 Related Work

Perspective distortion is a pervasive issue that significantly impacts the performance of various downstream tasks in computer vision. Perspective distortion arises from the camera's relative positioning to objects in the scene, causing apparent deformation in the captured image. This distortion results from the camera position, depth, intrinsic parameters (e.g., focal length, lens distortion), and extrinsic parameters (e.g., rotation, translation). These elements collectively affect the projection of 3D scenes onto 2D planes, impacting semantic interpretation and local geometry [20]. Accurately estimating these parameters for correcting perspective distortion is challenging and remains a critical barrier to developing robust computer vision methods. Another work [6] proposes a model that adapts 3D human pose estimation in videos to arbitrary camera distortions using meta-learning and a novel synthetic data generation technique.

Earlier studies on distortion correction primarily focused on rectifying perspective distortion, with limited emphasis on the robustness of computer vision applications. These correction methods typically transform computer vision tasks into two-stage processes: initially rectifying the distortion and subsequently engaging in task-specific learning. GeoNet [18] employs convolutional neural networks (CNNs) to predict distortion flow in images without prior knowledge of the distortion type. Perspective Crop Layers (PCL) [27] use perspective crop layers within CNNs to correct perspective distortions for 3D pose estimation. A cascaded deep structure network [21] addresses wide-angle portrait distortions without requiring calibrated camera parameters. Another study [29] predicts per-pixel displacement for face portrait undistortion. PerspectiveNet [15], and ParamNet [15] predict perspective fields and derive camera methods for camera calibration, respectively. Methods for fisheye image distortion correction have also been developed [25,26].

Earlier works have advanced computer vision tasks under perspective distortion. [24] introduced a method to correct perspective distortions in human mesh reconstruction from images with varying focal lengths. [16] proposed a framework for estimating camera parameters in the wild to improve 3D human pose and shape estimation. However, these methods rely on an inefficient two-step process and do not focus on robust representations for computer vision tasks.

Recently, the Möbius Transform for Mitigating Perspective Distortions (MPD) [5] introduced, emphasizing the impact of perspective distortion in real-world computer vision applications such as object recognition and detection. MPD utilizes Möbius transformations, a type of conformal mapping, to synthesize perspective distortion without estimating camera parameters. This method has shown the effectiveness of Möbius transformations in creating controlled distortions that mimic perspective distortion. While the Möbius transform effectively mimics perspective distortion, it presents significant challenges. MPD requires precise tuning of four interrelated complex parameters (a, b, c, d), which are not easily interpretable. This complexity makes the process sub-optimal and time-intensive. Furthermore, the Möbius transform inherently involves a series of complex arithmetic operations, specifically complex multiplications and divi-

sions, which contribute to its computational intensity. These operations require substantial computational resources, thereby limiting the efficiency of the transformation. Additionally, maintaining the stability of the Möbius transform is particularly challenging near singularities, where the condition $cz + d = 0$ leads to undefined behavior. These complexities highlight the need for more efficient and robust methods to synthesize perspective distortion, which underscores the need for continued research in this area.

3 Problem Formulation and Proposed Method

We begin by formalizing perspective distortion and reviewing the Möbius transform from MPD [5], highlighting its challenges. Next, we introduce our novel method, Log Conformal Maps (LCM), and provide proof of its non-linearity and conformity, establishing LCM as a viable candidate for mitigating perspective distortion. Furthermore, we demonstrate the efficiency of LCM by proving theoretically that LCM, with fewer parameters, can approximate the Möbius transform from MPD for mimicking perspective distortion.

3.1 Mathematical Foundation of Perspective Distortion

Perspective distortion arises from the projection of three-dimensional (3D) scenes onto a two-dimensional (2D) plane, typically described by the perspective projection model. This projection is inherently non-linear and non-conformal, causing significant visual distortions, especially when objects are viewed from angles other than the perpendicular.Perspective projection can be expressed as:

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

where (X, Y, Z) are the coordinates of a point in 3D space, (x', y') are the coordinates of the projected point on the 2D plane, f is the focal length, and w' is a scaling factor given by $w' = Z$.

The 2D coordinates (x, y) are obtained by normalizing with w':

$$x = \frac{x'}{w'} = \frac{fX}{Z}, \quad y = \frac{y'}{w'} = \frac{fY}{Z}$$

This non-linear relationship between the 3D coordinates and their 2D projections causes perspective distortion, where objects farther from the camera appear smaller and parallel lines seem to converge. Additionally, this projection does not preserve angles (non-conformal), further contributing to the distortion. Affine transformations, which include translations, rotations, scaling, and shearing, can be represented as linear transformations of the form:

where **A** is a matrix and **b** is a translation vector. However, affine transformations cannot model the non-linear and non-conformal nature of perspective distortion, as they preserve parallelism and ratios of distances along parallel lines, which are not preserved in perspective projection.

Given these limitations, non-linear transformations Möbius transform from MPD [5] and the proposed Log Conformal Maps, are required to accurately model and mitigate the effects of perspective distortion in computer vision applications.

3.2 Challenges of the Möbius Transform

The Möbius transformation, also known as a bilinear or fractional linear transformation, is a conformal mapping that preserves angles. It is defined as:

$$\Phi(z) = \frac{az + b}{cz + d} \tag{1}$$

where a, b, c, d are complex numbers and $ad - bc \neq 0$. This transformation maps the extended complex plane (including the point at infinity) onto itself. The MPD method utilizes the Möbius transformation to synthesize perspective distortions in images. This process involves mapping image coordinates to the complex plane, applying the Möbius transformation, and then mapping the transformed coordinates back to the image plane.

sub-optimal Despite its effectiveness, the Möbius transform poses several challenges. The transformation requires careful tuning of four interdependent complex parameters (a, b, c, d), each ranging between 0 and 1, creating a parameter space

$$\mathcal{P} = \{(a, b, c, d) \in \mathbb{C}^4 \mid ad - bc \neq 0\}.$$

This space grows exponentially because each parameter can take on numerous values, leading to n^4 possible combinations if discretized into n steps. The influence of parameters on each other means a change in one parameter significantly affects the others, complicating the optimization process. Determining the parameters, either heuristically or through a learning algorithm, is particularly challenging due to the vast and complex search space required to accurately mimic perspective distortion.

compute intensive The fractional aspect of the Möbius transformation, involving complex division, adds significant computational complexity. Calculating the real and imaginary parts of both the numerator and denominator, followed by their division, requires multiple operations per pixel:

$$\mathrm{Re}(\Phi(z)) = \frac{\mathrm{Re}(az + b) \cdot \mathrm{Re}(cz + d) + \mathrm{Im}(az + b) \cdot \mathrm{Im}(cz + d)}{(\mathrm{Re}(cz + d))^2 + (\mathrm{Im}(cz + d))^2} \tag{2}$$

$$\mathrm{Im}(\Phi(z)) = \frac{\mathrm{Im}(az+b)\cdot\mathrm{Re}(cz+d) - \mathrm{Re}(az+b)\cdot\mathrm{Im}(cz+d)}{(\mathrm{Re}(cz+d))^2 + (\mathrm{Im}(cz+d))^2} \tag{3}$$

These operations, repeated for each pixel, impose a substantial computational burden, especially for high-resolution images. Additionally, near singularities where $cz + d \approx 0$, the transformation becomes unstable, leading to very large or undefined values.

3.3 Proposed Method - Log Conformal Maps (LCM)

To address the challenges posed by the Möbius transform, we propose the Log Conformal Maps (LCM) method. LCM leverages the logarithmic function to approximate perspective distortions with fewer parameters and reduced computational complexity. With flip and rotation operations, LCM achieves all four perspective-distorted views (left, right, top, bottom) similar to the MPD [5]. The Log Conformal Transform is defined as:

$$\Psi(z) = \log(kz + c) \tag{4}$$

where k and c are complex numbers. The logarithmic function is non-linear yet conformal, preserving angles and providing a smooth distortion. LCM outlined in Algorithm 1. Proof on LCM non-linearity and conformality in sec. 3.4.

Algorithm 1 LCM Method for Perspective Distortion

Require: Input image I, width parameters k, and constant parameters c, rotation angles $\theta \in \{0°, 90°, 270°\}$, flip
Ensure: Transformed image I_{LCM} representing perspective distortion
$I_\theta \leftarrow \mathrm{rotate}(I, \theta) + flip$ ▷ Rotate image by θ degrees and flip based on view
for each pixel coordinate (x, y) in I_θ **do**
  $z \leftarrow x + iy$ ▷ Map pixel coordinates to complex vector
  $z_{\log} \leftarrow \log(kz + c)$ ▷ Apply Log Conformal Transform
  $x_{\log} + iy_{\log} \leftarrow z_{\log}$ ▷ Real and imaginary parts as transformed coordinates
  $(x_d, y_d) \leftarrow \mathrm{round}(x_{\log}), \mathrm{round}(y_{\log})$ ▷ Discretize to nearest pixel values
  $I_{LCM}(x_d, y_d) \leftarrow I_\theta(x, y)$ ▷ Assign pixel value to transformed coordinates
end for
$I_{LCM} \leftarrow \mathrm{rotate}(I_{LCM}, \theta) + flip$ ▷ Rotate and flip as post processing
$I_{LCM} \leftarrow \mathrm{Padding}(I_{LCM})$ ▷ Padding for smoother transformed output
return I_{LCM}

The Log Conformal Maps (LCM) method applies a logarithmic transform $\Psi(z) = \log(kz + c)$ to complex coordinates $z = x + iy$, providing non-linear and conformal mapping. Optionally, the method uses simple rotation and flip operations with transform to obtain all four perspective-distorted views: left, right, top, and bottom. This method simplifies parameter tuning with only the width parameter (k, reducing computational complexity as the logarithmic function avoids the intensive arithmetic of fractions required by the Möbius transform.

3.4 Non-linearity and Conformality of Log Conformal Maps

Theorem 1. *The Log Conformal Map (LCM) defined by $\Psi(z) = \log(kz + c)$, where k and c are complex numbers and z is a complex variable, is both non-linear and conformal.*

Proof. To establish that $\Psi(z) = \log(kz+c)$ is non-linear and conformal, we need to demonstrate two properties:
1. **Non-linearity**: The function $\Psi(z)$ do not satisfy the superposition principle.
2. **Conformality**: The function $\Psi(z)$ is holomorphic with a non-zero derivative, ensuring angle preservation.

Non-linearity A function $f(z)$ is non-linear if it does not satisfy the superposition principle:

$$\Psi(\alpha z_1 + \beta z_2) \neq \alpha\Psi(z_1) + \beta\Psi(z_2)$$

for complex numbers z_1, z_2 and scalars α, β. Consider two complex numbers z_1 and z_2:

$$\Psi(z_1 + z_2) = \log(k(z_1 + z_2) + c)$$

However,

$$\Psi(z_1) + \Psi(z_2) = \log(kz_1 + c) + \log(kz_2 + c) = \log((kz_1 + c)(kz_2 + c))$$

Certianly, $\log(k(z_1 + z_2) + c) \neq \log((kz_1 + c)(kz_2 + c))$. Therefore, $\Psi(z)$ is non-linear.

Conformality A function $f(z)$ is conformal if it preserves angles locally, which is ensured if the function is holomorphic (complex differentiable) with a non-zero derivative.
1. **Holomorphicity**:

$$\Psi(z) = \log(kz + c)$$

To show $\Psi(z)$ is holomorphic, we need to demonstrate it is complex differentiable. The derivative of $\Psi(z)$ is:

$$\Psi'(z) = \frac{d}{dz}\log(kz + c) = \frac{k}{kz + c}$$

2. **Non-zero Derivative**: For $\Psi(z)$ to be conformal, its derivative must be non-zero. Since $k \neq 0$ and $kz + c \neq 0$ for all $z \in \mathbb{C}$, the derivative $\Psi'(z) = \frac{k}{kz+c}$ is non-zero.

Therefore, since $\Psi(z)$ is holomorphic with a non-zero derivative, it is conformal and preserves angles locally.

Combining these results, we conclude that the Log Conformal Map (LCM), $\Psi(z) = \log(kz + c)$, is both non-linear and conformal.

4 Results and Discussions

We trained ResNet50 [11] for supervised learning using the protocol outlined in [19], and SimCLR [4] for self-supervised contrastive learning on the ImageNet [7] dataset. Additionally, the self-supervised method DINO [3] was trained following the authors' specified protocol. The term "Standard ResNet50" refers to the ResNet50 model trained on ImageNet as guided by [19], while "DINO" refers to the original works of SimCLR [4] and DINO [3]. We evaluated our methods on three public datasets: Imagenet-PD [5], explicitly introduced to benchmark images distorted by perspective distortion, and other perspective distortion-affected benchmarks, ImageNet-E [17] and Imagenet-X [14]. The results of these evaluations are presented in this section.

4.1 ImageNet-PD

ImageNet-PD [5] was introduced to benchmark robustness to perspective distortion. This dataset is derived from ImageNet classes and consists of four subsets, each corresponding to a different direction—left, right, top, and bottom—mimicking perspective distortion.

Supervised learning In Table 1, we provide results for ImageNet-PD. The hyper-parameter p represents the probability of applying LCM during the training process. From Table 1, it is evident that when training performance with standard protocol [19], there is a significant drop in accuracy. However, LCM adapted training results in a substantial boost in accuracy across all subsets of the ImageNet-PD dataset. From Figure 2, we observe that a fully supervised

Table 1: LCM trained on supervised learning and evaluated on **ImageNet-PD subsets** and original ImageNet. The probability P of applying LCM in model fine-tuning. results for probability P=0 denotes standard training and reported from MPD [5]. PD - Perspective Distorted.

P	Original ImageNet		Top-view (PD-T)		Bottom-view (PD-B)		Left-view (PD-L)		Right-view (PD-R)	
	Top1	Top 5	Top1	Top 5	Top1	Top 5	Top1	Top 5	Top1	Top 5
0.0	76.13±0.04	92.86±0.01	63.37±0.06	83.61±0.02	61.15±0.04	81.86±0.01	65.20±0.03	85.13±0.03	65.84±0.06	85.64±0.02
0.2	75.96±0.02	92.30±0.04	71.94±0.01	90.00±0.02	71.88±0.03	90.74±0.03	71.92±0.02	90.78±0.02	71.95±0.02	90.95±0.02
0.4	76.08±0.02	92.98±0.04	72.44±0.02	90.40±0.02	72.00±0.03	91.02±0.02	72.88±0.04	91.20±0.03	72.58±0.02	91.22±0.02
0.6	76.18±0.04	92.80±0.04	72.68±0.03	90.52±0.02	72.04±0.03	90.88±0.03	73.10±0.03	91.48±0.03	72.76±0.04	91.05±0.03
0.8	76.22±0.03	92.96±0.03	72.90±0.04	90.58±0.03	71.95±0.02	91.02±0.02	73.22±0.02	91.10±0.04	72.83±0.03	91.16±0.03
1.0	74.30±0.02	91.10±0.02	70.98±0.03	89.90±0.02	71.57±0.02	90.86±0.03	71.92±0.03	91.00±0.02	72.10±0.03	90.30±0.03

LCM adapted ImageNet trained model shows higher accuracy than standard ImageNet trained ResNet50 [19] and retains similar performance with highly compute-intensive MPD [5]. We also compare the LCM with other popular data augmentation methods in Table 2 on ImageNet-PD, where LCM demonstrates a significant performance lead over other methods in supervised approach.

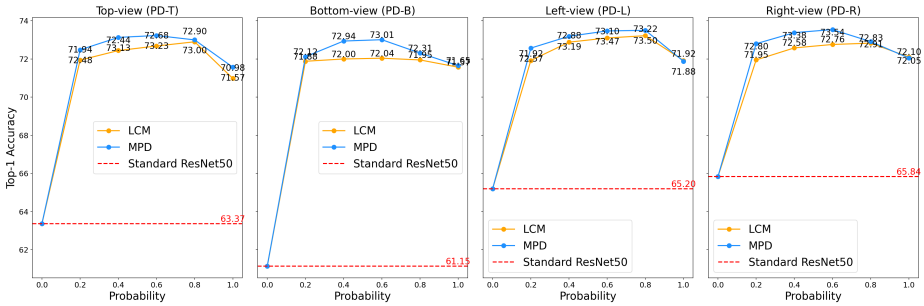


Fig. 2: Comparison of LCM with MPD method [5] across probability values. Red line shows standard ResNet50. LCM outperform ResNet50 model trained on ImageNet and matches the performance with MPD. MPD results from [5].

Table 2: Comparisons with other augmentation methods.

Method	Original ImageNet	ImageNet PD			
		Top-view (PD-T)	Bottom-view (PD-B)	Left-view (PD-L)	Right-view (PD-R)
	Top1	Top1	Top1	Top1	Top1
standard ResNet50	76.13	63.37	61.15	65.20	65.84
+Mixup [28]	77.46	65.46	66.79	68.02	68.43
+Cutout [8]	77.08	64.27	62.04	65.45	65.61
+AugMix [12]	77.53	64.12	62.90	65.95	66.49
+Pixmix [13]	77.37	65.52	64.76	67.26	67.56
LCM	76.22	**72.90**	**71.95**	**73.22**	**72.83**
LCM + SimCLR	76.60	**73.50**	**73.40**	**73.44**	**73.60**

Self-supervised Representation learning We investigate the robustness of LCM in the SimCLR self-supervised learning method, where pretraining is performed with LCM added as an augmentation with a probability of 0.8. Fine-tuning is then conducted following the protocol outlined in Section 4.1. Results for the self-supervised representation are provided in Table 3. As shown in Figure 3, LCM-trained ImageNet weights, when fully fine-tuned, outperform the standard ImageNet training and demonstrate competitive performance compared to the highly compute-intensive MPD method. Similar to the supervised approach, self-supervised trained LCM models also showcase significant performance improvement compared to other popular data augmentations (Table 2).

Table 3: LCM trained with SimCLR[4] self-supervised pretraining with probability 0.8 and evaluated on **ImageNet-PD subsets** and original ImageNet. The probability P of applying LCM in model fine-tuning stage. Results for probability P denotes standard training from MPD [5]. PD - Perspective Distorted.

P	Original ImageNet		Top-view (PD-T)		Bottom-view (PD-B)		Left-view (PD-L)		Right-view (PD-R)	
	Top1	Top 5	Top1	Top 5	Top1	Top 5	Top1	Top 5	Top1	Top 5
0.2	76.44±0.03	93.42±0.01	72.25±0.03	90.90±0.01	72.36±0.04	90.82±0.03	72.36±0.03	91.26±0.01	72.60±0.02	91.22±0.02
0.4	76.60±0.04	93.12±0.03	73.12±0.02	91.20±0.02	73.12±0.02	91.60±0.03	73.48±0.02	91.60±0.02	73.42±0.03	91.80±0.02
0.6	76.08±0.02	93.20±0.04	73.10±0.04	91.80±0.04	73.20±0.04	91.40±0.03	73.42±0.04	91.55±0.01	73.52±0.02	91.62±0.04
0.8	76.60±0.02	92.96±0.04	73.50±0.02	91.22±0.02	73.40±0.03	91.76±0.04	73.44±0.03	91.78±0.03	73.60±0.01	91.66±0.04
1.0	74.38±0.02	91.35±0.02	71.45±0.02	90.20±0.02	71.55±0.02	90.40±0.03	71.88±0.02	90.56±0.04	72.00±0.04	90.64±0.03

We also investigate the linear evaluation performance of self-supervised methods SimCLR [4] and DINO [3]. Similar to SimCLR, DINO is adapted with LCM as an augmentation, and pretraining is performed, followed by linear evalua-

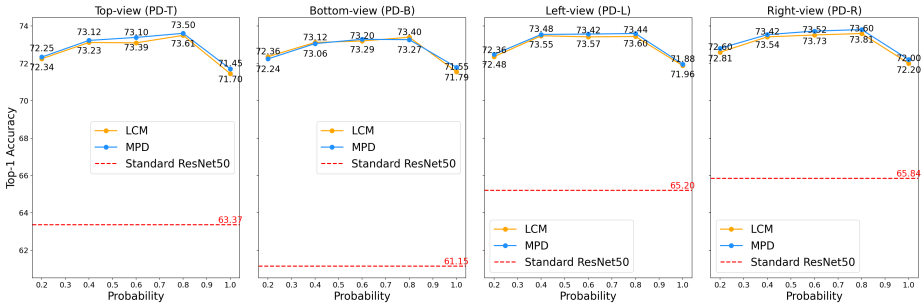


Fig. 3: Comparison of self-supervised trained LCM+SimCLR with MPD+SimCLR method [5] across probability values. Red line shows standard ResNet50. LCM outperform ResNet50 model trained on ImageNet and matches the performance with MPD. MPD results are reported from [5].

Table 4: Linear evaluation of SimCLR [4] contrastive self-supervised models on **ImageNet-PD sub-sets**. Self-supervised pretraining was performed on ImageNet dataset with batch size of 512 on linear learning rate for 100 epochs, refer SimCLR[4]. The probability of LCM is set to 0.8 for self-supervised pre-training and linear evaluation. ResNet50 is backbone. results for standard SimCLR and SimcLR+MPD are reported from MPD [5].

Model	Original ImageNet	Top-view (PD-T)	Bottom-view (PD-B)	Left-view (PD-L)	Right-view (PD-R)
standard SimCLR	60.14	34.22	32.87	36.33	36.89
SimCLR+MPD	60.02±0.03	50.63±0.03	49.65±0.03	50.41±0.03	50.64±0.05
SimCLR+LCM	**60.30±0.02**	**51.04±0.03**	**50.00±0.03**	49.88±0.04	49.78±0.02

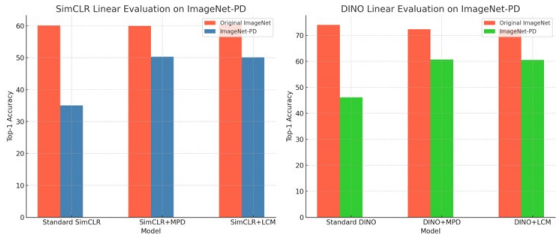


Fig. 4: Comparison of linear evaluation performance on self-supervised trained models. (Left): original SimCLR [4], LCM+SimCLR and MPD+SimCLR[5] on original ImageNet and ImageNet-PD subsets. Performance on ImageNet-PD is average over all four subsets. (Right): Comparing original DINO [3], LCM+DINO and MPD+DINO [5] on original ImageNet and ImageNet-PD subsets. Performance on ImageNet-PD is average over all subsets.

tion. LCM outperforms the standard original SimCLR [4] and DINO [3] when integrated and linearly evaluated (see Figure 4). Detailed results are shown in Tables 4 and 5. Notably, while LCM-adapted self-supervised models outperform ImageNet-PD subsets, they retain performance on the original ImageNet.

Table 5: Linear evaluation of knowledge distillation based self-supervised method DINO [3] on **ImageNet-PD sub-sets**. Self-supervised pretraining was performed on ImageNet dataset with batch size of 512 on linear learning rate for 100 epochs, refer DINO[3]. The probability of LCM is set to 0.8 for self-supervised pre-training and linear evaluation. Results for standard DINO and DINO+MPD are reported from MPD [5]. ViT-small transformer [9] is common backbone.

Model	Original ImageNet	Top-view (PD-T)	Bottom-view (PD-B)	Left-view (PD-L)	Right-view (PD-R)
standard DINO	74.00	46.31	46.05	46.15	45.98
DINO+MPD	72.36±0.01	60.72±0.02	60.86±0.02	60.63±0.02	60.58±0.02
DINO+LCM	**72.42±0.02**	**61.10±0.02**	**61.06±0.03**	59.85±0.03	60.02±0.01

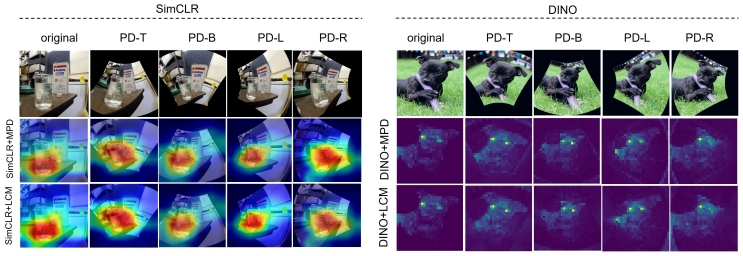


Fig. 5: Activation maps: (Left) *beaker* example (n02815834) in ImageNet-PD subsets comparing MPD and LCM in self-supervised learning method SimCLR [4]. (Right) *Staffordshire bullterrier* example (n02093256) in ImageNet-PD subsets comparing MPD and LCM in self-supervised learning method DINO [3]. Reported result for MPD is used and same example were used for comparison.

4.2 Qualitative Analysis

In this section, we present qualitative results in Figure 5 to assess the robustness of our proposed formulation. We have plotted the Grad-CAM for the LCM-trained SimCLR method and attention maps for the LCM-trained DINO method, comparing these with the Möbius and LCM methods. The Grad-CAM results for the LCM-trained SimCLR method show clearly defined attention regions, indicating effective feature localization. Similarly, the attention maps for the LCM-trained DINO method demonstrate precise focus areas. These visualizations highlight that LCM, with fewer parameters, produces comparative qualitative outcomes to MPD [5].

4.3 ImageNet-E

ImageNet-Editing, commonly known as ImageNet-E [17], is a comprehensive dataset for benchmarking robustness against various object attributes, including perspective distortion. We extensively evaluated the LCM-trained models, both supervised and self-supervised, and compared their performance with other methods. The comparison are presented in Figure 6 and detailed results are in Table 9 in suppl. material. The main observation is a significant boost in accuracy when LCM is incorporated into the training procedure, benefiting both background and object size changes in the dataset's images. Figure 6

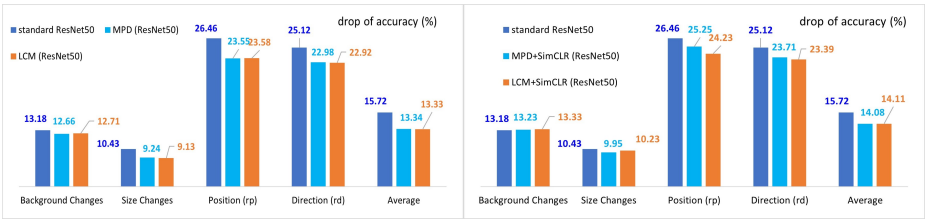


Fig. 6: ImageNet-E: Drop of Top1 accuracy under background changes, size changes, random position (rp), random direction (rd), and average over 11 subsets. Lower is better. The mean is reported for size and background-related subsets in ImageNet-E[17]. *Average* reports the mean of all subsets. (Left): compares LCM with MPD [5] in supervised approach. (Right): Compares LCM with MPD [5] in SimCLR self-supervised learning approach.

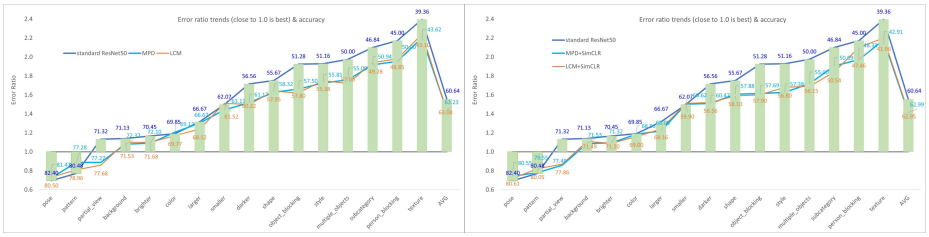


Fig. 7: ImageNet-X[14]: Compares error ratio and accuracy of 16 factors for standard ResNet50 model with LCM and MPD. Results for standard ResNet50 ImageNet trained model and MPD trained model are reported from MPD [5]. Error ratio close to 1.0 shows highest robustness [14]. (Left): Compares LCM with standard ResNet50 and MPD in supervised learning. (Right): Compares LCM with standard ResNet50 and MPD in SimCLR [4] self-supervised learning.

compares our proposed LCM-based technique with the existing Möbius-based MPD [5] method, demonstrating that the LCM-trained models outperform standard trained models overall the subsets and offers competitive performance compared with MPD [5] with fewer parameters and less compute-intensive, showing resilient for robustness.

4.4 ImageNet-X

ImageNet-X [14] contains human annotations identifying failure types in the widely-used ImageNet dataset. These annotations label distinguishing object factors such as pose, size, color, lighting, occlusions, and co-occurrences for each image in the validation set and a random subset of 12,000 training samples. We evaluated LCM-trained ImageNet models on ImageNet-X, and our findings are reported in Figure 7 and detailed results are in Table 10 in suppl. material. Figure 7 illustrates a reduction in the error ratio towards the optimal value of 1.0, decreasing from 1.55 to 1.43 in fully supervised experiments. A similar trend is observed for the self-supervised method, indicating increased robustness.

4.5 LCM Compute Efficiency

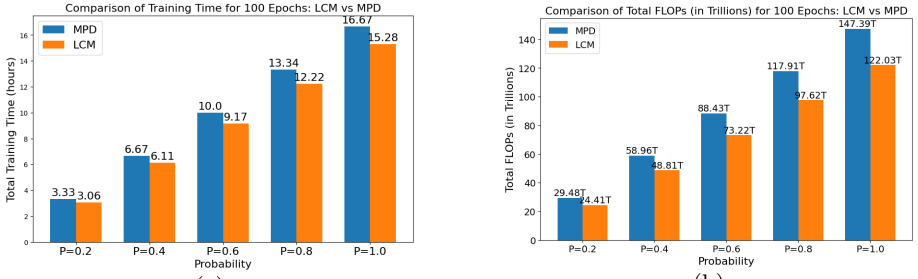
To demonstrate the computational efficiency of the proposed LCM method over the existing Möbius Transform-based MPD method [5], we performed a FLOP analysis, with the details presented in Table 6. Notably, LCM requires fewer floating point operations per second (FLOPs) than the MPD method. The reduced FLOPs are primarily due to LCM's less complex arithmetic operations compared to Möbius transformations. Our benchmarking of the compute time for image transformation operations using LCM and MPD on a CPU-only setup further underscores the efficiency of the LCM method. The results show that MPD requires 0.24 seconds per image per transform, while LCM, with its streamlined approach, needs only 0.22 seconds per image per transform when using a single-core CPU. Table 7 shows the compute efficiency of LCM over MPD.

Table 6: Computation efficiency of LCM

Operation	FLOPs per Pixel	MPD	LCM
Meshgrid Creation	2	2 × H × W	2 × H × W
Complex Arithmetic for Transformation	14	14 × H × W	0
Logarithmic Mapping	4	0	4 × H × W
Scaling and Conversion to Image Coordinates	6	0	6 × H × W
Grid Sampling (Bilinear Interpolation)	7	7 × H × W	7 × H × W
Total FLOPs		23 × H × W	19 × H × W

Table 7: Compute efficiency and power consumption between MPD and LCM.

Metric	MPD	LCM
Time/epoch	0.1667 hours	0.1528 hours
Complete training	16.67 hours	15.28 hours
Saved hours	-	**1.39 hrs (8.33%)**
Power/epoch	400.08 kWh	366.72 kWh
Total power	4000.8 kWh	3667.2 kWh
Saved power	-	**333.6 kWh**



(a) (b)

Fig. 8: Comparison of LCM and MPD for (a) training time efficiency and (b) FLOPs across probability ranges.

To compare the compute efficiency of LCM and MPD methods, we performed a detailed compute analysis with our pretraining experiments using SimCLR self-supervised pretraining on a ResNet50 model with the ImageNet dataset. The model trains for 100 epochs with an input size of 224x224 and a batch size of 512, distributed across 8 NVIDIA H100 GPUs. Both methods show nearly identical per-image processing times (LCM: 0.22s, MPD: 0.24s). LCM demonstrates significant efficiency, saving 1.39 hours (8% increase) and 333.6 kWh over

100 epochs (Table 7 and Fig. 8a). LCM reduces FLOPs upto 17% compared to MPD (Fig. 8b). If method-specific time is discounted, DINO will show a similar trend in compute efficiency.

5 Person Re-identification

Following MPD [5], we integrated LCM with CLIP-ReIdent [10], a CLIP-based contrastive image-image pre-training for person re-identification on DeepSportRadar dataset [23] having video frames of varying poses and camera angles of players (Fig. 9). LCM-Clip-ReIdent models outperform original Clip-ReIdent and improve mean-average precision (mAP) across backbone ViT-L-14 and ResNet50x16 (Table 8) for both approaches, with and without re-ranking.

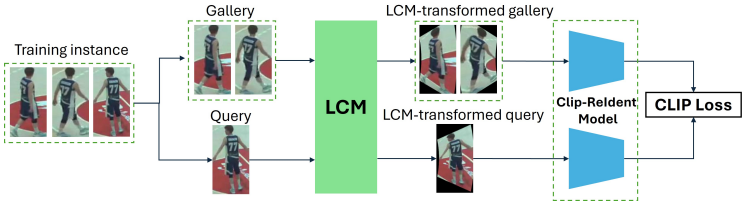


Fig. 9: LCM-Clip-ReIdent: LCM transforms the input gallery and query during training of Clip-ReIdent for person re-id. LCM probability is set to 0.1.

Table 8: LCM improves Clip-ReIdent model.

Method	Encoder	mAP (w/o Re-ranking)	mAP (with Re-ranking)
Baseline	ViT-L-14	72.70	-
Clip-ReIdent		96.90	98.20
MPD-Clip-ReIdent		97.02	98.30
LCM-Clip-ReIdent		**97.16**	**98.42**
Clip-ReIdent	ResNet50x16	88.50	94.90
MPD-Clip-ReIdent		**91.95**	97.50
LCM-Clip-ReIdent		91.90	**97.62**

6 Conclusion and Future Work

In this paper, we present the Log Conformal Maps (LCM) transform to efficiently handle perspective distortion, enhancing model robustness. Our analysis, backed by experiments, shows LCM's effectiveness in modeling perspective distortion. LCM-adapted models in both supervised and self-supervised settings surpass standard models and achieve state-of-the-art performance. Given its relevance to tasks like crowd counting and object detection, future directions include exploring LCM's application to diverse computer vision tasks and its computational advantages. We aim to inspire further research into using logarithmic conformal mapping for addressing geometric challenges in computer vision.

Acknowledgment: The authors thank Sumit Rakesh, Luleå University of Technology, for his support with the Lotty Bruzelius cluster. We also thank the National Supercomputer Centre at Linköping University for the Berzelius supercomputing, supported by the Knut and Alice Wallenberg Foundation.

References

1. Arnold, D.N., Rogness, J.P.: Möbius transformations revealed. Notices of the American Mathematical Society **55**(10), 1226–1231 (2008)
2. Bottger, J., Balzer, M., Deussen, O.: Complex logarithmic views for small details in large contexts. IEEE Transactions on Visualization and Computer Graphics **12**(5), 845–852 (2006)
3. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9650–9660 (2021)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of the International Conference on Machine Learning. pp. 1597–1607. PMLR (2020)
5. Chhipa, P.C., Chippa, M.S., De, K., Saini, R., Liwicki, M., Shah, M.: Möbius transform for mitigating perspective distortions in representation learning. In: Proceedings of the European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024. (2024)
6. Cho, H., Cho, Y., Yu, J., Kim, J.: Camera distortion-aware 3d human pose estimation in video with optimization-based meta-learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11169–11178 (2021)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
8. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (2020)
10. Habel, K., Deuser, F., Oswald, N.: Clip-reident: Contrastive training for player re-identification. In: Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports. pp. 129–135 (2022)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
12. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. In: International Conference on Learning Representations
13. Hendrycks, D., Zou, A., Mazeika, M., Tang, L., Li, B., Song, D., Steinhardt, J.: Pixmix: Dreamlike pictures comprehensively improve safety measures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16783–16792 (2022)
14. Idrissi, B.Y., Bouchacourt, D., Balestriero, R., Evtimov, I., Hazirbas, C., Ballas, N., Vincent, P., Drozdzal, M., Lopez-Paz, D., Ibrahim, M.: Imagenet-x: Understanding model mistakes with factor of variation annotations. Proceedings of the International Conference on Learning Representations (2023)
15. Jin, L., Zhang, J., Hold-Geoffroy, Y., Wang, O., Blackburn-Matzen, K., Sticha, M., Fouhey, D.F.: Perspective fields for single image camera calibration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17307–17316 (2023)

16. Kocabas, M., Huang, C.H.P., Tesch, J., Müller, L., Hilliges, O., Black, M.J.: Spec: Seeing people in the wild with an estimated camera. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 11035–11045 (2021)
17. Li, X., Chen, Y., Zhu, Y., Wang, S., Zhang, R., Xue, H.: Imagenet-e: Benchmarking neural network robustness via attribute editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20371–20381 (2023)
18. Li, X., Zhang, B., Sander, P.V., Liao, J.: Blind geometric distortion correction on images through deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4855–4864 (2019)
19. PyTorch: Vision: Datasets, transforms and models specific to computer vision (2023), `https://github.com/pytorch/vision/tree/main/references/classification`, original repository for PyTorch Vision Reference Scripts, ACCESSED: August, 1, 2023
20. Rahman, T., Krouglicof, N.: An efficient camera calibration technique offering robustness and accuracy over a wide range of lens distortion. IEEE Transactions on Image Processing **21**(2), 626–637 (2011)
21. Tan, J., Zhao, S., Xiong, P., Liu, J., Fan, H., Liu, S.: Practical wide-angle portraits correction with deep structured models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3498–3506 (2021)
22. Vadapally, B.K., Rahman, Z.u.: Image registration using conformal log polar mapping. In: Visual Information Processing XVIII. vol. 7341, pp. 118–128. SPIE (2009)
23. Van Zandycke, G., Somers, V., Istasse, M., Don, C.D., Zambrano, D.: Deepsportradar-v1: Computer vision dataset for sports understanding with high quality annotations. In: Proceedings of the 5th International ACM Workshop on Multimedia Content Analysis in Sports. pp. 1–8 (2022)
24. Wang, W., Ge, Y., Mei, H., Cai, Z., Sun, Q., Wang, Y., Shen, C., Yang, L., Komura, T.: Zolly: Zoom focal length correctly for perspective-distorted human mesh reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3925–3935 (2023)
25. Yang, S., Lin, C., Liao, K., Zhao, Y.: Innovating real fisheye image correction with dual diffusion architecture. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12699–12708 (2023)
26. Yin, X., Wang, X., Yu, J., Zhang, M., Fua, P., Tao, D.: Fisheyerecnet: A multi-context collaborative deep network for fisheye image rectification. In: Proceedings of the European Conference on Computer Vision. pp. 469–484 (2018)
27. Yu, F., Salzmann, M., Fua, P., Rhodin, H.: Pcls: Geometry-aware neural reconstruction of 3d pose with perspective crop layers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9064–9073 (2021)
28. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
29. Zhao, Y., Huang, Z., Li, T., Chen, W., LeGendre, C., Ren, X., Shapiro, A., Li, H.: Learning perspective undistortion of portraits. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7849–7859 (2019)

7 Supplementary Material

7.1 ImageNet-E

In Table 9, we report both the drop in accuracy and the absolute accuracy for the fully supervised ResNet-50 network and the self-supervised SimCLR [4] method.

Table 9: ImageNet-E: (a) Drop of Top1 accuracy (Lower is better) and (b) Absolute Accuracy (Higher is better) under background changes, size changes, random position (rp), random direction (rd) categories. Results for ResNet50 are reported from MPD [5].

(a) Drop of Accuracy

Models	Original	Background changes					Size changes				Position	Direction
		Inver	λ = -20	λ = 20	λ = 20-adv	Random	Full	0.10	0.08	0.05	rp	rd
standard ResNet50	92.69	1.97	7.30	13.35	29.92	13.34	2.71	7.25	10.51	21.26	26.46	25.12
LCM (ResNet50)	92.60	1.72	6.70	11.55	29.95	13.64	3.20	6.08	8.7	18.54	23.58	22.92
LCM+SimCLR (ResNet50)	92.55	1.40	7.36	12.23	31.53	14.15	2.96	6.84	10.57	20.55	24.23	23.39

(b) Absolute Accuracy

Models	Original	Background changes					Size changes				Position	Direction
		Inver	λ = -20	λ = 20	λ = 20-adv	Random	Full	0.10	0.08	0.05	rp	rd
standard ResNet50	92.69	90.72	85.39	79.34	62.77	79.35	89.98	85.44	82.18	71.43	66.23	67.57
LCM (ResNet50)	92.60	90.88	85.90	81.05	62.65	78.96	89.40	86.52	83.90	74.06	69.02	69.68
LCM+SimCLR (ResNet50)	92.55	91.15	85.19	80.32	61.02	78.40	89.59	85.71	81.98	72.00	68.32	69.16

7.2 ImageNet-X

Following the detailed results in Table 10, it is evident that LCM-trained fully supervised ImageNet weights and self-supervised ImageNet weights offer better accuracies than the standard weights. Notably, LCM-trained weights demonstrate better or more competitive performance when compared to MPD-trained ImageNet weights.

Table 10: ImageNet-X: error ratios comparisons for MPD and LCM. Error ration close to 1.0 is ideal robustness for each factor. LCM demonstrate consistent robustness compared to standard ResNet50 model and matches the performance with MPD in supervised and self-supervised approaches.

Factor	Supervised			Self-supervsied	
	Standard ResNet50	MPD	LCM	MPD+SimCLR	LCM+SimCLR
pose	0.70	0.72	0.73	0.74	0.72
pattern	0.77	0.89	0.81	0.78	0.82
partial_view	1.13	0.89	0.86	0.86	0.87
background	1.14	1.08	1.10	1.09	1.11
brighter	1.17	1.09	1.10	1.09	1.09
color	1.19	1.21	1.17	1.18	1.16
larger	1.32	1.30	1.24	1.22	1.24
smaller	1.50	1.44	1.51	1.50	1.51
darker	1.72	1.52	1.50	1.51	1.52
shape	1.75	1.63	1.64	1.61	1.59
object_blocking	1.92	1.66	1.62	1.61	1.60
style	1.93	1.73	1.75	1.63	1.68
multiple_objects	1.97	1.76	1.72	1.72	1.69
subcategory	2.10	1.92	1.94	1.90	1.84
person_blocking	2.17	1.95	1.98	1.97	2.10
texture	2.39	2.20	2.24	2.18	2.20
Average	1.55	1.44	**1.43**	1.41	1.42