# Lightweight Optimal-Transport Harmonization on Edge Devices

**Maria Larchenko**[1], **Dmitry Guskov**[2, 3], **Alexander Lobashev**[2], **Georgy Derevyanko**[1]

[1] Magicly AI, Dubai, UAE
[2]Glam AI, San Francisco, USA
[3]McGill University, Montreal, Canada
{mariia.larchenko, guskov01dmitry, lobashevalexander, georgy.derevyanko}@gmail.com

## Abstract

Color harmonization adjusts the colors of an inserted object so that it perceptually matches the surrounding image, resulting in a seamless composite. The harmonization problem naturally arises in augmented reality (AR), yet harmonization algorithms are not currently integrated into AR pipelines because real-time solutions are scarce. In this work, we address color harmonization for AR by proposing a lightweight approach that supports on-device inference. For this, we leverage classical optimal transport theory by training a compact encoder to predict the Monge-Kantorovich transport map. We benchmark our MKL-Harmonizer algorithm against state-of-the-art methods and demonstrate that for real composite AR images our method achieves the best aggregated score. We release our dedicated AR dataset of composite images with pixel-accurate masks and data-gathering toolkit to support further data acquisition by researchers.

**Code** —
https://github.com/maria-larchenko/mkl-harmonizer

## Introduction

Composite images are created by pasting a foreground object onto a background image under a binary mask. Although spatial alignment may be perfect, the inserted region usually looks out of place because it was captured under different illumination, camera response, or post-processing pipeline. Image harmonization modifies the pasted region so that its appearance becomes perceptually consistent with its surroundings, yielding a harmonized image. Existing harmonization methods are predominantly offline and assume desktop-class resources, limiting their adoption in interactive applications (Niu et al. 2021).

Early and influential works adopted encoder-decoder architectures, such as U-Net (Ronneberger, Fischer, and Brox 2015), to perform a dense, pixel-to-pixel mapping of the foreground. Models like DoveNet (Cong et al. 2020), RainNet (Ling et al. 2021), or IntrinsicHarmony (Guo et al. 2021) were able to learn rich semantic representations, leading to significant improvements over classical techniques. However, dense prediction models share a fundamental limitation in their computational and memory requirements. Consequently, research was largely confined to low-resolution inputs, typically 256x256 pixels.



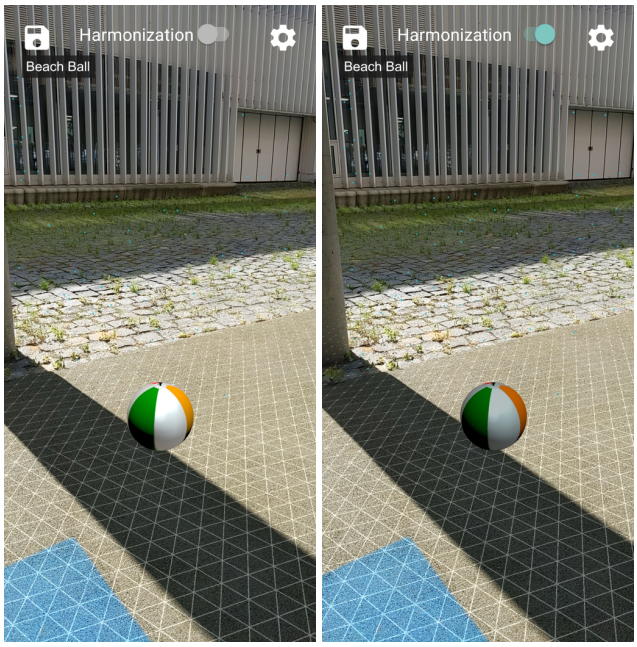


Figure 1: Harmonization running on edge device.

To address these limitations, Harmonizer (Ke et al. 2022) regresses coefficients for filters like brightness and contrast, while DCCF (Xue et al. 2022) proposes a set of neural color filters whose application is controlled by a predicted coefficient map. More recent approaches have focused on learning the transformation itself. PCT-Net (Guerreiro, Nakazawa, and Stenger 2023) learns to predict the parameters for a smooth field of pixel-wise affine color transforms. A key contribution was to perform upsampling on the low-resolution parameter map rather than the image itself. INR-Harmonization (HINet) (Chen et al. 2023) introduced the use of an Implicit Neural Representation, where an encoder predicts the weights of a small MLP that maps a pixel's coordinate to its final harmonized color.

While deep learning has dominated recent literature, the principles of color transfer are rooted in classical statistical

methods. Recently, Optimal Transport (OT) theory has experienced a significant resurgence, particularly in the context of generative modeling (Liu, Gong, and Liu 2023).

Historically, OT provided a way to map one color distribution to another. For Gaussian approximation of source and target color distributions, optimal transport map has a closed-form linear solution, known as Monge-Kantorovich Linear map (MKL) (Pitié and Kokaram 2007). For the color transfer problem, the MKL filter yields perceptually coherent results. Later, Rabin et al. (Rabin, Delon, and Gousseau 2010) introduced a relaxed formulation of OT for adaptive color transfer and Bonneel et al. (Bonneel et al. 2013) applied OT principles to the complex task of video color grading. MKL filters first introduced in 2007 are still a strong competitor for the more recent color transfer algorithms (Larchenko et al. 2025).

To apply MKL color filter one has to calculate 12 parameters, derived from statistics of source and target color distributions. To the best of our knowledge, this approach was not previously tested for the color harmonization problem. Since the target color distribution is unknown for inserted object, our approach is to train a network to predict the 12 parameters of Monge-Kantorovich Linear transformation. The proposed solution is promisingly lightweight, fast and fits into the constraints of on-device augmented reality.

Our work makes the following contributions

- We propose MKL-Harmonizer, a novel lightweight solution for color harmonization, based on prediction of Monge-Kantorovich Linear maps.
- We built a dedicated tool to gather AR-specific composite images and masks in the wild and have collected a set of 327 images for human evaluation.
- We demonstrate that our approach achieves superior performance in terms of aggregated speed-quality metric.
- We deploy our solution on edge devices and measure its real-time performance.

## Background

**Augmented Reality** AR systems render virtual content into a live camera stream at video rate. Here, composition and harmonization happen every frame: the renderer generates a synthetic object that must match the photometric properties of the camera feed on resource-constrained hardware (mobile GPUs, XR headsets, embedded devices). Despite its importance for realism, advanced color harmonization is missing from today's mainstream AR tool-chains like ARKit, ARCore, Meta Spark, Snap Lens Studio. The main barriers are latency and mobile compute limits. Instead, these platforms rely on light estimation mechanisms:

- Main direction of light and it's intensity
- Environmental cube maps (Greene 1986)
- Spherical harmonic lighting (Ramamoorthi and Hanrahan 2001)
- Global adjustments such as exposure or white-balance.

Because of computational restrictions color harmonization for AR objects remains an under-studied area. However, this field demonstrates its own unique challenges.

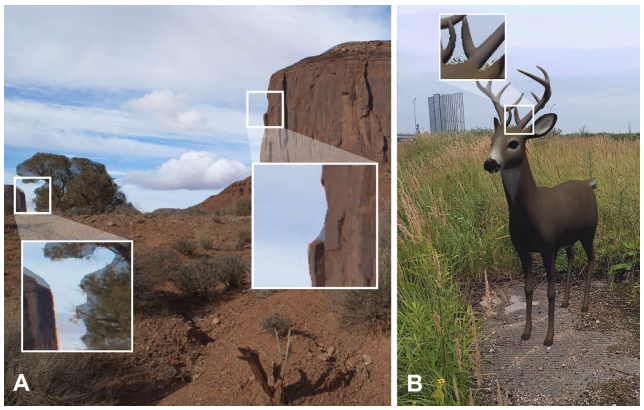


Figure 2: Exposure bias in image harmonization. Augmented image from iHarmony4 training partition. (A) Small regions near the boundary of the mask contains an information about unharmonized background and can inadvertently teach the harmonization network to over-rely on this specific information. (B) For object inserted from a 3D engine the mask is "pixel-perfect".

A typical training data for harmonization comprises of real photos or rendered scenes which then are augmented to produce synthetic composite images, i.e. every augmented image has a ground true solution. For AR composite images, with a real image on the background and virtual foreground objects, it is impossible to produce training data in the same way. Thus, even training on both real and virtual scenes leads to out-of-distribution input during inference.

Secondly, AR composites do not suffer from the imperfect masks common in photo-editing datasets, as the rendering engine provides a pixel-perfect mask for every virtual object. On one hand, this simplifies harmonization by removing visible mask artifacts model should care about. On the other hand, models trained on real composites may struggle with inference on AR data due to the exposure bias.

**Exposure Bias in Image Harmonization** Harmonization algorithms are typically trained on datasets of composite images where foreground object masks are generated either manually or through segmentation algorithms. A key characteristic of these masks, such as those in the widely-used iHarmony4 dataset, is that they often include background pixels near the object's boundary, as shown in Fig. 2 A. This "pixel-imperfect" nature can provide the model with crucial information about the ground truth harmonized background.

The presence of these boundary pixels can inadvertently teach the harmonization network to over-rely on this specific information. The model learns to predict the harmonization filter by comparing the "leaked" background pixels within the mask to the visible unharmonized background. This comparison is an effective shortcut, as these two background regions are far more similar in content than the foreground object and the background are.

This reliance creates a training-inference mismatch, a problem also known as the exposure bias. When the model is later applied to an object rendered by a 3D engine or cut

from another image, the mask is typically "pixel-perfect" and contains no leaked background information. Since the model was trained to depend on this boundary information, its performance can degrade significantly when it is absent.

**Optimal Transport** Color distributions can be modeled with continuous or discrete probability density function in RGB space. The problem of color harmonization can be seen as predicting a new color distribution for the pasted object based on the scene. That is, we want to find a new color distribution for the object as if it was in the scene originally.

We denote the original color density as $\pi_0$ and the predicted color density as $\pi_1$, with $im$ representing a scene image. The random variables $X_0 \sim \pi_0$ and $X_1 \sim \pi_1$ represent pixels sampled from their respective distributions. The harmonization problem means finding a deterministic transport map $T_{im} : \mathbb{R}^3 \to \mathbb{R}^3$ such that $T_y(X_0) = X_1$, i.e.

$$\pi_0(x) = \pi_1(T_{im}(x)) \left|\det J_{T_{im}}(x)\right|, \tag{1}$$

where $J_T(x)$ is the Jacobian of $T$ taken at point $x$.

**Monge's Optimal Transportation** To select a unique map from the many that could satisfy the mass-preserving property in Eq. 1, we introduce a cost function. With a standard quadratic cost, $c(x, y) = \|x - y\|^2$, the Monge problem seeks the optimal deterministic map $T^*_{im}$ that minimizes the total expected cost of transforming the source distribution to the target:

$$\text{Cost}\left[T_{im}\right] = \mathbb{E}\big(\left\|X_1 - X_0\right\|^2\big) \tag{2}$$

$$= \int_{\mathcal{X}_0} (T_{im}(x) - x)^2 \pi_0(x) dx. \tag{3}$$

For continuous density functions with finite second moments, a unique solution to this problem is guaranteed to exist (Villani 2009).

When both the source and target color distributions are approximated as multivariate Gaussians, i.e., $X_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$ and $X_1 \sim \mathcal{N}(\mu_1, \Sigma_1)$, this optimal transport map has a well-known closed-form linear solution, the Monge-Kantorovich Linear (MKL) filter (Pitié and Kokaram 2007). The optimal map $T^*_{im}$ is given by:

$$T^*_{im}(x) = \mu_1 + A(x - \mu_0), \tag{4}$$

where the linear transformation matrix $A$ is computed as:

$$A = \Sigma_0^{-1/2} \left(\Sigma_0^{1/2} \Sigma_1 \Sigma_0^{1/2}\right)^{1/2} \Sigma_0^{-1/2}. \tag{5}$$

The transformation is fully characterized by the means $\mu_0, \mu_1$ and covariance matrices $\Sigma_0, \Sigma_1$. Our method, therefore, involves training a neural network to predict these target statistics $(\mu_1, \Sigma_1)$ based on the input scene, from which the MKL filter can be directly computed.

## Theoretical Analysis

This section investigates the theoretical conditions under which a linear MKL map can effectively approximate a complex, non-linear color harmonization task. We provide a formal justification by bounding the approximation error, demonstrating that the MKL filter is effective when the true color transformation is smooth (i.e., has a small Lipschitz constant) and color distributions are not pathologically located at the extreme boundaries of the color gamut.

**Model Assumptions and Preliminaries** Let the color space be the unit cube $\mathcal{X} = [0, 1]^3$. Let the unharmonized and target color distributions, $\pi_0$ and $\pi_1$, be probability measures supported on $\mathcal{X}$. We impose the following regularity conditions for our analysis.

**Assumption 1 (Map Regularity).** The true optimal transport map, $T^* : \mathcal{X} \to \mathcal{X}$, exists and is $L$-Lipschitz continuous for a constant $L < \infty$. The existence of such a map for a general case is non-trivial and is guaranteed under strong conditions on the measures, such as uniform log-concavity of their densities (Caffarelli 1992).

**Assumption 2 (Distribution Regularity).** The source distribution $\pi_0$ has mean $\mu_0$ and a non-singular covariance matrix $\Sigma_0$, which means that $\Sigma_0^{1/2}$ needed for the MKL map is well-defined. All expectations $\mathbb{E}[\cdot]$ are taken with respect to $X_0 \sim \pi_0$.

Our algorithm approximates $T^*$ with the MKL map, $T_{\text{MKL}}(x) = \mu_1 + A(x - \mu_0)$, which is the optimal map for transporting between Gaussian surrogates $\mathcal{N}(\mu_0, \Sigma_0)$ and $\mathcal{N}(\mu_1, \Sigma_1)$ (Peyré and Cuturi 2019). Since the MKL map could potentially map part of the density outside of the unit cube, we apply clipping operation $\hat{T}_{\text{MKL}} := \Pi_{\mathcal{X}} \circ T_{\text{MKL}}$, where $\Pi_{\mathcal{X}}$ is the Euclidean projection onto $\mathcal{X}$, i.e. clipping transformed colors to the valid $[0, 1]^3$ gamut. We proof that the clipping operation coincides with the Euclidean projection in Lemma 1.

**Lemma 1** (Clipping equals Euclidean projection)**.** *Let the* clipping operator $\text{clip} : \mathbb{R}^d \to [0, 1]^d$ *be defined componentwise by*

$$(\text{clip}(z))_j = \min\{1, \max\{0, z_j\}\}, \qquad j = 1, \ldots, d. \tag{6}$$

*Then for every $z \in \mathbb{R}^d$ the vector $y = \text{clip}(z)$ is the unique Euclidean projection of $z$ onto the cube $\mathcal{X} = [0, 1]^d$; that is,* $\text{clip}(z) = \Pi_{\mathcal{X}}(z)$.

Let us note that any projection operator is 1-Lipschitz since the distance between any two points after projection is never greater than their distance before projection. We seek to bound the expected squared error $\mathcal{E} := \mathbb{E}[\|\hat{T}_{\text{MKL}}(X_0) - T^*(X_0)\|^2]$.

**Theorem 1** (Error Bound for L-Lipschitz Color Maps)**.** *Let Assumptions 1 and 2 hold. The total error $\mathcal{E}$ is bounded as:*

$$\mathcal{E} \leq 2\mathcal{E}_{clip} + 2\mathcal{E}_{lin}, \tag{7}$$

*where the clipping error is $\mathcal{E}_{clip} := \mathbb{E}[\|T_{MKL}(X_0) - \hat{T}_{MKL}(X_0)\|^2]$, and the linearity error, $\mathcal{E}_{lin}$, is bounded by:*

$$\mathcal{E}_{lin} \leq 2B^2 + 2(\|A\|_{op} + L)^2 \cdot \text{tr}(\Sigma_0). \tag{8}$$

*Here, $B = |\mu_1 - T^*(\mu_0)|$ is a bias term, $||A||_{op}$ is the spectral norm of the MKL matrix, i.e. its largest singular value, which depends only on source and target distribution covariances, and $\text{tr}(\Sigma_0)$ is the trace of the source covariance matrix.*

*Proof Sketch.* Insert and subtract $T_{\mathrm{MKL}}(X_0)$ inside the norm, then use $\|u+v\|^2 \le 2\|u\|^2 + 2\|v\|^2$ to obtain $\mathcal{E} \le 2\mathcal{E}_{\text{clip}} + 2\mathcal{E}_{\text{lin}}$. To bound $\mathcal{E}_{\text{lin}}$, add and subtract $T^*(\pi_0)$, invoke the $L$-Lipschitz property of $T^*$ together with $\|A\|_{\text{op}}$ to control $\|T_{\mathrm{MKL}}(x) - T^*(x)\|$ by a bias term $B$ plus a scaled deviation from the mean. Squaring, taking expectations, and using $\mathbb{E}\|X_0 - \mu_0\|^2 = \mathrm{tr}(\Sigma_0)$ yields the claimed bound. □

This theorem bounds the approximation error, linking it to the Lipschitz constant $L$ of the true optimal transport map and the tail probability $\mathcal{E}_{clip} \le d \cdot \mathbb{P}[T_{\mathrm{MKL}}(X_0) \notin \mathcal{X}]$. The latter bound holds due to the following Lemma 2.

**Lemma 2** (Tail–probability bound for the clipping error)**.** *Let $\Pi_{\mathcal{X}} = \mathrm{clip}(\cdot)$ be the Euclidean projection onto $\mathcal{X} = [0,1]^d$, defined as*

$$(\mathrm{clip}(z))_j := \min\{1,\ \max\{0,\ z_j\}\}, \tag{9}$$

*for $j = 1, \dots, d,\ z \in \mathbb{R}^d$ and define*

$$\mathcal{E}_{clip} := \mathbb{E}\big[\|Z - \Pi_{\mathcal{X}} Z\|^2\big], \qquad Z \in \mathbb{R}^d. \tag{10}$$

*Then*

$$\mathcal{E}_{clip} \le d\,\mathbb{P}[Z \notin \mathcal{X}], \tag{11}$$

*and for our application with $Z = T_{\mathrm{MKL}}(X_0)$ and $d = 3$,*

$$\mathcal{E}_{clip} \le 3\,\mathbb{P}\big[T_{\mathrm{MKL}}(X_0) \notin \mathcal{X}\big]. \tag{11$'$}$$

In practice, color harmonization is a smooth process that maps adjacent colors to nearby ones, ensuring a small Lipschitz constant. This, combined with the fact that color distributions in natural images are rarely concentrated at the gamut boundaries, makes the clipping error negligible and justifies our linear approximation.

However, this is not the case when the harmonization task considers dark objects. Their color distribution is concentrated around a corner of $\mathcal{X}$. In our experiments we indeed observe that processing of too dark objects often leads to implausible results.

## Method

**Ground-Truth Filter Generation** To verify whether our theoretical assumptions hold (i.e. a simple linear filter is powerful enough for image harmonization) we first have computed standard set of iHarmony metrics for so-called *Ideal Linear OT filter*. This is not a predictive model, but rather a theoretical ceiling: for each image in the iHarmony dataset, we compute and store the exact MKL transform (Eq. 4) that optimally maps the color distribution of the augmented unharmonized distribution to its ground-truth. This ideal linear filter achieves a Mean Squared Error (MSE) $\sim 7.0$. which is quite low compared to the state-of-the-art results for this benchmark, see Tab. 1. High performance of linear MKL filters suggests that real harmonization maps are often Lipschitz and are mostly not concentrated near the gamut boundaries. Therefore, we use these pre-computed vectors as the primary supervisory signal during training.

**Loss function** At this point, we have two options. The first one is to train the encoder network Model$(\cdot)$ to predict statistics $\mu_1, \Sigma_1$ of the target harmonized distribution $\pi_1$. Since the input $\mu_0, \Sigma_0$ is known, the resulting filter is computed straightforwardly following Eq. 4. The second option is to directly predict $A$ given by Eq. 5 and shift $S$

$$S = \mu_1 - A\mu_0 \tag{12}$$

In our experiments we test both objectives for predicting tuples $[\mu_1, \Sigma_1]$ or $[A, S]$. We find that the latter one is more robust to the prediction inaccuracies and yields lower loss overall.

$$L_{labels} = ||\text{Model}(im) - [A, S]||_1 \tag{13}$$

In addition to this standard labels MSE loss, we found it beneficial to include the content L1 per-pixel loss for $[A', S']$ predicted by the model

$$L_{content} = ||M * X_0 - M * (X_0 \cdot A' + S')||_1 \tag{14}$$

where $M$ is binary mask, $X_0$ is flattened composite image pixels, $*$ denotes element-wise product and $\cdot$ is matrix product.

Since our problem is ill-posed, there is no unique MKL filter that satisfies the harmonization objective. Minimizing the expected $L_2$ error would therefore force the estimator to produce the unique arithmetic mean of all possible MKL solutions, which may not correspond to any actual filter. In contrast, we employ the $L_1$ loss, which is minimized by any valid solution and is, in a sense, less restrictive. The $L_1$ loss allows the network to converge to a sharper solution rather than the overly smoothed compromise imposed by $L_2$. Metrics for $L_{labels}$ with $L_1$ and $L_2$ losses are present in the Tab. 1.

The question arises, why not to stick to the $L_{content}$ only? In practice, when training with just $L_{content}$, model learns filters that are close to identity transformation. The possible explanation of this behavior is that in the absence of MKL guiding signal the model experiences mode collapse similar to the case described in PCT-Net training, where authors introduce contrastive loss to avoid filter collapse. So our final loss takes the form:

$$L_{total} = L_{labels} + \alpha L_{content} \tag{15}$$

We discuss parameters choice in the experiments section.

## Experiments and Metrics

**Datasets and Metrics** We train our model on iHarmony4 dataset (Cong et al. 2020), which contains synthesized composite images, foreground masks of composite images and corresponding real images.

For evaluation we use approach, standard for the color harmonization area: mean squared error (MSE), peak signal-to-noise ratio (PSNR) and foreground MSE (fMSE) calculated on the iHarmony test set in 256x256 resolution.

However, since we initially aimed to study harmonization in augmented reality, the central place in our evaluation is taken by AR-specific data.

**ARCore Evaluation Set** AR images are real composite images (i.e. produced without augmentation and thus they

Figure 3: The 3D objects used in our experiments feature different sizes and textures.

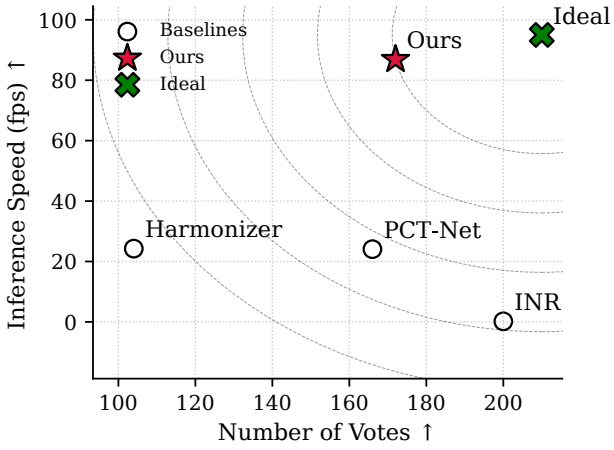


Figure 4: Mean opinion score versus inference speed calculated for images with 1080x2204 resolution from ARCore data. Data was processed on NVIDIA RTX 4060Ti GPU.

have no ground truth), making the standard metrics inapplicable. Moreover, AR data is scarce by itself. While one can find many options with fully rendered scenes like HVIDIT (Guo et al. 2021) or Hypersim (Roberts et al. 2021), there are no collections of relevant AR images with per-pixel masks available in the open source.

For this reason, we modify sample ARCore [1] application to turn it into basic data-gathering tool and collect the first small-scaled dataset of this kind. Currently the set contains 327 pairs of composite-mask images, captured in the wild, as shown in Fig. 1. It features various indoor and outdoor scenes, different day time, weather and lighting conditions. 3D objects used in our experiments are collected from open source and converted into the proper format. Selected object does not follow the common style and varies in sizes in textures as shown in Fig. 3. Please refer to the Supplementary for model credits and licenses. The rendering pipeline of application we used for data accumulation and model testing is depicted in Fig. 5. Crucially, all masks are obtained directly from the rendering engine.

**Baselines** We compare our method against three harmonization models described in the introduction section: Harmonizer (Ke et al. 2022), PCT-Net (Guerreiro, Nakazawa, and Stenger 2023), INR-Harmonization (Chen et al. 2023). Also, for iHarmony4 dataset, we include a comparison with classical color transfer between foreground and background pixels (Reinhard et al. 2001). All baseline models are evaluated using MSE, PSNR, and fMSE metrics, results are shown in Tab. 1.

**User Study on ARCore Data** For the user study, participants were shown the results of four harmonization methods – Harmonizer, PCT-Net, INR-Harmonization and our MKL encoder – applied to the same composite AR image and asked the question: "Which image is more natural and realistic?". Each image set was shuffled. We gathered 20 participants, each of whom graded around 30 sets of these 4-way comparisons as demonstrated in Fig. 9, resulting in total 642 grades (see Fig. 6)

**Encoder** We use an EfficientNet-B0 (Tan and Le 2019), also used in PCT-Net and Harmonizer, chosen for its efficiency on mobile hardware. We modify its first convolutional layer to accept a 4-channel input (RGB + mask), allowing it to process both color and spatial information simultaneously. The network is trained on $256 \times 256$ inputs and its final regression head outputs a 12-dimensional vector corresponding to the parameters of the approximate MKL filter.

**Training** The model is trained for 210 epochs on an NVIDIA RTX 4090 using the Adam optimizer (Kingma and Ba 2014) with a learning rate starting at $1e{-}3$ and decreasing at 50 epochs to $1e{-}4$ and at 110 epochs to $5e{-}5$ with a batch size of 64. We employ a hybrid loss function to ensure perceptual quality:

$$L_{total} = L_{labels} + \alpha \cdot L_{content} \tag{16}$$

We set the content loss weight $\alpha = 10$. This hybrid loss is critical for preventing the model from collapsing to a simple identity transformation as discussed in the Method section.

**Perceptual Evaluation Results** Given that image harmonization is an ill-posed problem, and in light of the exposure bias, we argue that perceptual evaluation by human observers is more insightful.

The results of our user study, summarized in Fig. 6, reveal two key insights. First, our method is rated perceptually on par with the leading baselines on real AR data. It proves that for pixel-perfect masks, where edges inconsistency is not an issue, simpler solution can be effective.

Second, this exposes the weakness of relying on MSE based metrics; while the INR model has notably lower MSE score compared with PCT-Net, human observers rate its perceptual quality much higher, making MOS a more reliable metric. We suggest that the difference between the human scores and the metric in Tab. 1 may arise due to the training-inference mismatch discussed in the background section. The qualitative comparison is given in Fig.7.

Furthermore, Tab. 2 and Fig. 4 illustrate the essential trade-off between perceptual quality (MOS) and inference

[1] AR platform for Android developed and supported by Google

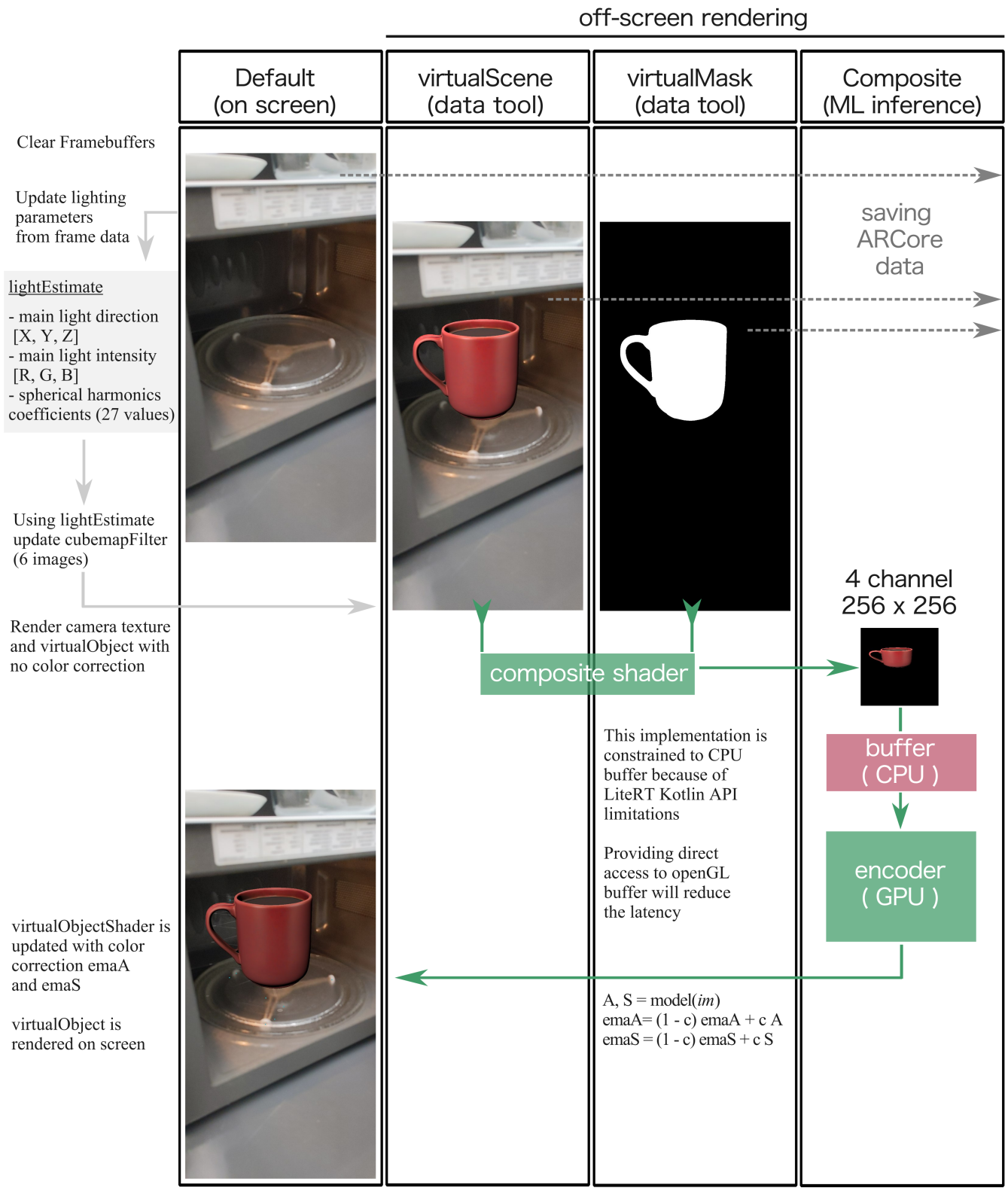


Figure 5: The general scheme of main rendering loop demonstrates four openGL Framebuffers. Default one performs on-screen rendering, others are used for off-screen rendering and auxiliary tasks, such as data collection. Limitations of LiteRT Next Kotlin introduces computational overhead due to copying model input through the CPU. Masks are rendered concurrently with foreground objects, obtained directly from the rendering engine and are saved on user's demand.

| Method | MSE | PSNR | fMSE |
|---|---|---|---|
| Ideal Linear OT | 7.6 ± 0.2 | 43.6 ± 0.1 | 45.9 ± 0.9 |
| PCT-Net | 29.1 ± 0.9 | 38.0 ± 0.1 | 201 ± 4 |
| Harmonizer | 40.1 ± 1.2 | 36.6 ± 0.1 | 258 ± 5 |
| Ours $L_1$ | 65.0 ± 1.6 | 34.1 ± 0.1 | 438 ± 7 |
| Ours $L_2$ | 66.3 ± 1.7 | 33.9 ± 0.1 | 451 ± 7 |
| INR | 67.2 ± 1.8 | 35.3 ± 0.1 | 392 ± 7 |
| CT | 284 ± 6.9 | 27.5 ± 0.1 | 1836 ± 23 |
| Unharmonized | 182 ± 5 | 31 ± 0.1 | 984 ± 17 |

Table 1: Quantitative results on the iHarmony4 256x256 dataset. Ideal Linear OT filter represents reference values calculated based on ground true images. Even though these values are not achievable in practice, they prove linear filters may be capable enough.

| Method | 256×256 | 512×512 | 1024×2048 | 4096×4096 |
|---|---|---|---|---|
| Ours | 175.01 | 166.76 | 137.21 | 40.85 |
| DoveNet | 123.39 | – | – | – |
| PCT-Net | 104.57 | 98.65 | 63.74 | 11.84 |
| Harmonizer | 95.01 | 89.82 | 47.63 | 7.45 |
| INR | 6.35 | 3.22 | 0.81 | 0.12 |

Table 2: Performance test on different resolutions measured in iterations per second. Experiment was carried out on RTX 4060Ti.

speed. Our method offers both the high perceptual score and the fastest performance.

**Inference on an Edge Device** We evaluate the on-device inference performance of our method using the Google Pixel 4a and Google Pixel 7. We handle model conversion via the LiteRT Next Kotlin API and run the model on each frame of the rendering loop. Frame rate performance is measured on devices before and after turning on online harmonization, resulting in 12 to 15 fps, see Fig. 8. However, our tested implementation makes two unnecessary passes thought CPU buffers (see Fig. 5). Implementing zero-copy routines could potentially double this framerate to 24 - 30 fps range (Google AI Edge 2025).

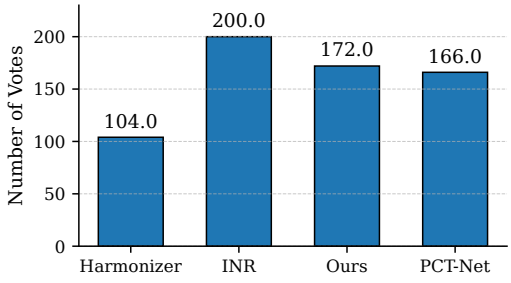


Figure 6: Mean opinion score on ARCore data. Results discussed in section ARCore Dataset and user Study.

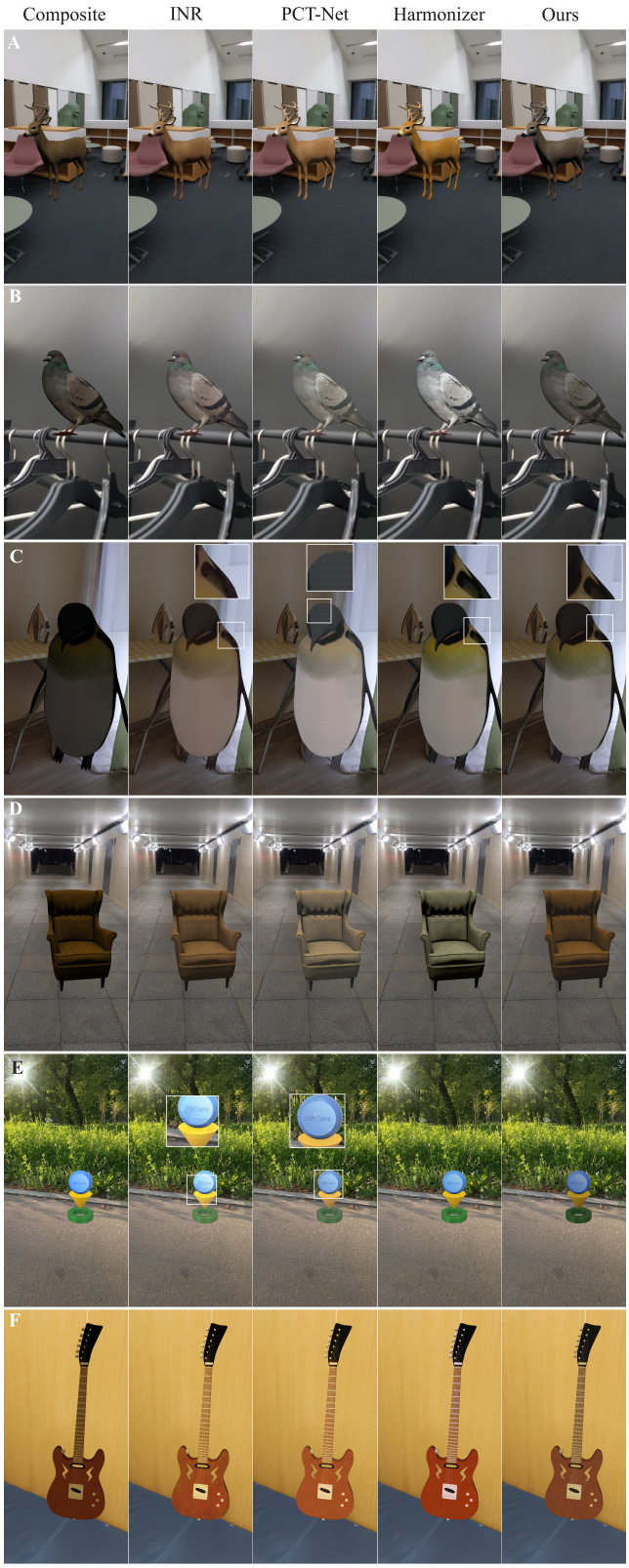


Figure 7: Qualitative comparison with baselines.

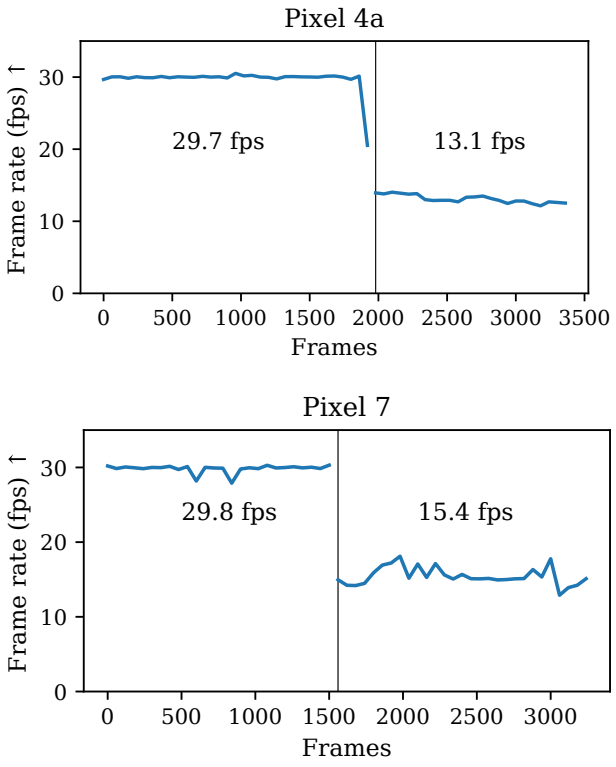


Figure 8: Frame rate profiles from the device before and after harmonization is turned on.

## Discussion and Limitations

For high-resolution images, we observe that dense prediction models often produce corrupted results, leading to coarse or degraded outputs. In contrast, Harmonizer and our method do not exhibit these defects, as they are filter-based methods and thus avoid upscaling-related artifacts. In Fig. 7C and E, one can observe coarse stripes in PCT-Net and pixelated areas resembling JPEG artifacts in INR. Notably, these issues are not present in low-resolution settings. However, all models exhibit certain biases in their predictions—for instance, Harmonizer tends to increase image brightness, while our predictions tend to be darker. Another limitation of our method is that it is not designed as a final solution for video harmonization. Since our model is not trained on video data, sequential predictions may vary significantly across frames. To mitigate this, we apply an exponential moving average, as illustrated in Fig. 5. However, video harmonization remains a challenging problem overall.

## Conclusion

In this work, we introduced a lightweight and efficient method for color harmonization tailored for real-time augmented reality applications. We framed the harmonization task as an optimal transport problem and grounded our method in classical OT theory by training an encoder to predict the parameters of a Monge-Kantorovich Linear filter. Furthermore, we provided a theoretical analysis that justifies the use of a linear map by bounding its approximation error, showing it is effective for the smooth color transforms typical of harmonization.

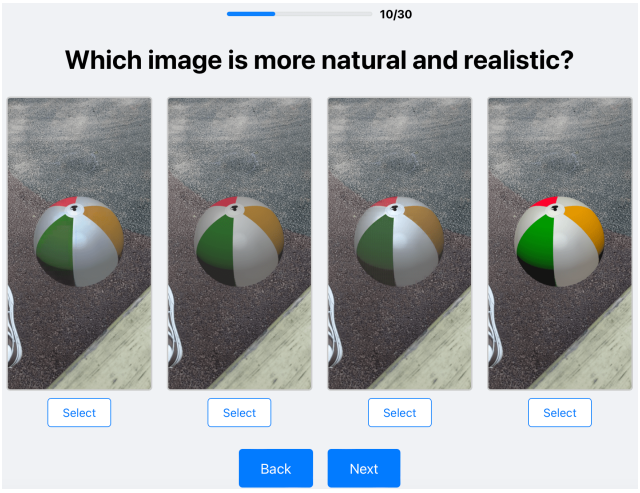


Figure 9: User interface of the labeling page.

Our central contribution is not only the model itself but also a pioneering analysis of AR-specific harmonization challenges. We critically evaluated how state-of-the-art methods are performed and identified a possible "exposure bias" in the standard iHarmony4 dataset, which may cause metrics like MSE to misrepresent perceptual quality. To address this, we created and introduced the ARCore dataset, a new benchmark with pixel-perfect masks for realistic AR evaluation.

## References

Bonneel, N.; Sunkavalli, K.; Paris, S.; and Pfister, H. 2013. Example-based video color grading. *ACM Trans. Graph.*, 32(4): 39–1.

Caffarelli, L. A. 1992. The regularity of mappings with a convex potential. *Journal of the American Mathematical Society*, 5(1): 99–104.

Chen, J.; Zhang, Y.; Zou, Z.; Chen, K.; and Shi, Z. 2023. Dense pixel-to-pixel harmonization via continuous image representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(5): 3876–3890.

Cong, W.; Zhang, J.; Niu, L.; Liu, L.; Ling, Z.; Li, W.; and Zhang, L. 2020. Dovenet: Deep image harmonization via domain verification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8394–8403.

Google AI Edge. 2025. LiteRT: Async Segmentation Sample - Performance. https://github.com/google-ai-edge/LiteRT/tree/main/litert/samples/async_segmentation#performance.

Greene, N. 1986. Environment mapping and other applications of world projections. *IEEE computer graphics and Applications*, 6(11): 21–29.

Guerreiro, J. J. A.; Nakazawa, M.; and Stenger, B. 2023. Pct-net: Full resolution image harmonization using pixel-wise color transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5917–5926.

Guo, Z.; Zheng, H.; Jiang, Y.; Gu, Z.; and Zheng, B. 2021. Intrinsic image harmonization. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 16367–16376.

Ke, Z.; Sun, C.; Zhu, L.; Xu, K.; and Lau, R. W. 2022. Harmonizer: Learning to perform white-box image and video harmonization. In *European conference on computer vision*, 690–706. Springer.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Larchenko, A.; Lobashev, A.; Guskov, D.; and Palyulin, V. V. 2025. Color Transfer with Modulated Flows. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 4464–4472.

Ling, J.; Xue, H.; Song, L.; Xie, R.; and Gu, X. 2021. Region-aware adaptive instance normalization for image harmonization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9361–9370.

Liu, X.; Gong, C.; and Liu, Q. 2023. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Niu, L.; Cong, W.; Liu, L.; Hong, Y.; Zhang, B.; Liang, J.; and Zhang, L. 2021. Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490*.

Peyré, G.; and Cuturi, M. 2019. Computational Optimal Transport. *Foundations and Trends® in Machine Learning*, 11(5-6): 355–607.

Pitié, F.; and Kokaram, A. 2007. The linear monge-kantorovitch linear colour mapping for example-based colour transfer. In *4th European conference on visual media production*, 1–9. IET.

Rabin, J.; Delon, J.; and Gousseau, Y. 2010. Regularization of transportation maps for color and contrast transfer. In *2010 IEEE International Conference on Image Processing*, 1933–1936. IEEE.

Ramamoorthi, R.; and Hanrahan, P. 2001. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 497–500.

Reinhard, E.; Adhikhmin, M.; Gooch, B.; and Shirley, P. 2001. Color Transfer between Images. *IEEE Computer Graphics and Applications*, 21(5): 34–41.

Roberts, M.; Ramapuram, J.; Ranjan, A.; Kumar, A.; Bautista, M. A.; Paczan, N.; Webb, R.; and Susskind, J. M. 2021. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. In *International Conference on Computer Vision (ICCV) 2021*.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, 6105–6114. PMLR.

Villani, C. 2009. *Optimal Transport: Old and New*. Springer.

Xue, B.; Ran, S.; Chen, Q.; Jia, R.; Zhao, B.; and Tang, X. 2022. Dccf: Deep comprehensible color filter learning framework for high-resolution image harmonization. In *European conference on computer vision*, 300–316. Springer.