IntrinsicEdit: Precise generative image manipulation in intrinsic space

LINJIE LYU, Max-Planck-Institute for Informatics, Saarland Informatics Campus, Germany and Adobe Research, UK
VALENTIN DESCHAINTRE, Adobe Research, UK
YANNICK HOLD-GEOFFROY, Adobe Research, Canada
MILOŠ HAŠAN, Adobe Research, USA
JAE SHIN YOON, Adobe Research, USA
THOMAS LEIMKÜHLER, Max-Planck-Institute for Informatics, Saarland Informatics Campus, Germany
CHRISTIAN THEOBALT, Max-Planck-Institute for Informatics, Saarland Informatics Campus, Germany
ILIYAN GEORGIEV, Adobe Research, UK

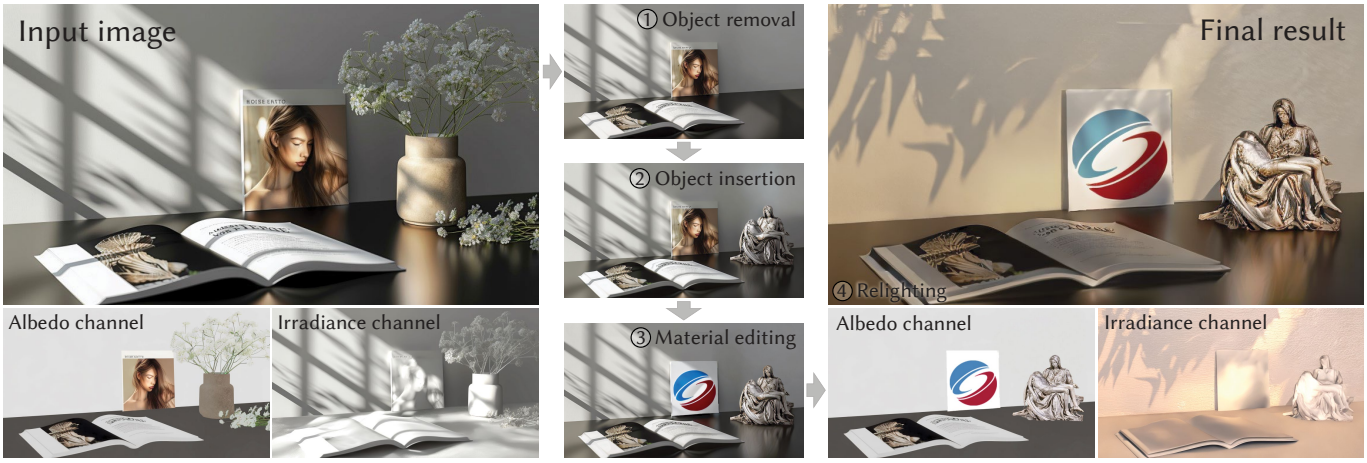

Fig. 1. We propose a generative framework for diverse image-editing tasks, where precise manipulations can be performed in an intrinsic-image space and global-illumination effects are subsequently resolved automatically. Here we show a progressive transformation of an input image: ① We first remove the flowers and the vase from the albedo channel and then ② insert a new object in that channel. ③ We replace the texture of another object before ④ relighting the scene using a new irradiance channel. After each intrinsic-channel manipulation, we can render a physically plausible result. No single prior method can perform all these edits and provide similar levels of precision and identity preservation while delivering comparable image quality.

Generative diffusion models have advanced image editing by delivering high-quality results through intuitive interfaces such as prompts, scribbles, and semantic drawing. However, these interfaces lack precise control, and associated editing methods often specialize in a single task. We introduce a versatile workflow for a range of editing tasks which operates in an intrinsic-image latent space, enabling semantic, local manipulation with pixel precision while automatically handling effects like reflections and shadows. We build on the RGB↔X diffusion framework and address its key deficiencies: the lack of identity preservation and the need to update multiple channels to achieve plausible results. We propose an edit-friendly diffusion inversion and prompt-embedding optimization to enable precise and efficient editing of only the relevant channels. Our method achieves identity preservation and resolves global illumination, without requiring task-specific model fine-tuning. We demonstrate state-of-the-art performance across a variety of tasks on complex images, including material adjustments, object insertion and removal, global relighting, and their combinations.

Code will be available at our project page: https://intrinsic-edit.github.io.

CCS Concepts: • **Computing methodologies** → **Image manipulation**.

Additional Key Words and Phrases: Image editing, intrinsic decomposition, diffusion models, identity preservation, realistic rendering

ACM Reference Format:
Linjie Lyu, Valentin Deschaintre, Yannick Hold-Geoffroy, Miloš Hašan, Jae Shin Yoon, Thomas Leimkühler, Christian Theobalt, and Iliyan Georgiev. 2025. IntrinsicEdit: Precise generative image manipulation in intrinsic space. *ACM Trans. Graph.* 44, 4 (August 2025), 13 pages. https://doi.org/10.1145/3731173

Authors' Contact Information: Linjie Lyu, Max-Planck-Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany and Adobe Research, London, UK, llyu@mpi-inf.mpg.de; Valentin Deschaintre, Adobe Research, London, UK, deschain@adobe.com; Yannick Hold-Geoffroy, Adobe Research, Quebec, Canada, holdgeof@adobe.com; Miloš Hašan, Adobe Research, San Jose, USA, mihasan@adobe.com; Jae Shin Yoon, Adobe Research, San Jose, USA, jaeyoon@adobe.com; Thomas Leimkühler, Max-Planck-Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany, thomas.leimkuehler@mpi-inf.mpg.de; Christian Theobalt, Max-Planck-Institute for Informatics, Saarland Informatics Campus, Saarbrücken, Germany, theobalt@mpi-inf.mpg.de; Iliyan Georgiev, Adobe Research, London, UK, igeorgiev@adobe.com.

ACM 1557-7368/2025/8-ART
https://doi.org/10.1145/3731173

1 Introduction

Image editing is a fundamental operation in the creative domain. Editing tasks range from subtle local corrections to more substantial modifications, such as altering the appearance and layout of objects or adjusting lighting. Achieving high-fidelity edits has traditionally required significant expertise and time. We propose a generative method that operates in an intrinsic-image space and significantly simplifies multiple non-trivial editing tasks: object insertion/removal, material manipulation, relighting.

Generative diffusion models [Ho et al. 2020; Rombach et al. 2022; Sohl-Dickstein et al. 2015] have recently revolutionized imaging, as researchers realized that such models are capable of not only generating new images from text prompts, but can be repurposed for inpainting and other non-trivial edits. Recent work has introduced intuitive interfaces based on prompting [Brooks et al. 2023; Sheynin et al. 2024], dragging [Shi et al. 2024; Wu et al. 2025], scribbling [Ding et al. 2024; Lee et al. 2024], or semantic drawing [Zhang et al. 2023b; Zhu et al. 2025]. Despite ease of use, such methods are typically limited to relatively high-level control, making it challenging to precisely define the desired edit while keeping the remaining image content intact. A partial solution to preserve identity is to mask the pixels that should remain unaffected [Avrahami et al. 2022]. However, this can be difficult when intricate, non-local effects such as shadows, reflections, and color bleeding need to be considered, as these typically have fuzzy boundaries that are hard to anticipate.

Methods that perform intrinsic image decomposition and resynthesis [Kocsis et al. 2024b; Luo et al. 2024; Zeng et al. 2024a] promise accurate control over the entire image by representing it through channels that encode per-pixel information about geometry, appearance, and lighting. The RGB↔X framework of Zeng et al. [2024a], illustrated in Fig. 2, envisions an editing pipeline where one (i) decomposes the input image into intrinsic channels using an RGB→X model, (ii) applies adjustments to these channels, and finally (iii) recomposes an edited image using a neural rendering X→RGB model. This approach is inspired by classical 3D workflows where geometry, appearance, and lighting are defined and manipulated independently, enabling precise, physically based editing.

However, the practical realization of this promising vision requires solving the challenges of (i) identity preservation and (ii) the need to edit multiple channels simultaneously. Indeed, unlike a complete 3D-scene representation, the intrinsic-image space encodes only a subset of the information required to perfectly reproduce the input image, leaving room for the neural renderer to sample from an entire distribution of images consistent with the intrinsic conditions, necessarily shifting identity. Furthermore, the intrinsic channels carry redundant information; removing or inserting an object by (say) editing the albedo channel requires non-trivial updates to other channels to render a faithfully edited result.

In this work, we address both the identity preservation and channel entanglement limitations of the RGB↔X framework to unlock its full image-editing potential. Furthermore, we do so exclusively with inference-time techniques, without requiring any further training. We first ensure that we can reconstruct the input image from the estimated intrinsic channels, which is necessary to preserve identity. We achieve this by performing exact inversion of the X→RGB model. The resulting noise vector may contain too much image-specific information baked in, which hinders editability. To avoid that, prior to inversion we optimize the originally unused X→RGB text-prompt embedding to absorb such information and encode it as a model condition. Second, to address the entanglement of intrinsic channels and gain the freedom to edit only the best-suited one(s) for the current task, we encode the remaining channels into the prompt embedding, at the same time as we optimize it for the aforementioned edit-friendly inversion. This channel-to-prompt transfer enables the model to preserve the non-edited properties of the input image more abstractly and flexibly than direct per-pixel specifications, while still allowing for precise, localized editing of the channel(s) of interest, making for a streamlined editing process.

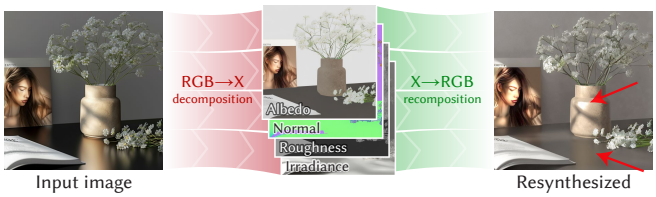

Fig. 2. **RGB↔X overview.** An RGB→X diffusion model decomposes a given image into intrinsic channels, while a complementary neural rendering X→RGB diffusion model composes channels into an image [Zeng et al. 2024a]. The complete image-to-image RGB→X→RGB pipeline promises semantic editing with pixel precision by manipulating the channels before recomposition. Unfortunately, the models' generative nature causes random identity shifts in the resynthesized image, and successful editing requires adjusting multiple entangled channels, hindering usability. We address both these issues to unlock the image-editing potential of RGB↔X.

Our technical advancements combine to enable a wide range of image editing tasks within a single framework. This framework features an interpretable latent space—the intrinsic images, which allows for pixel-level control through both traditional and modern image manipulation tools. Moreover, our method can achieve edits that other approaches struggle to perform well, such as pasting a texture onto an existing object, seamlessly integrating an inserted 3D object with specular reflections, or fully relighting a scene—as shown in Fig. 1, all while automatically resolving global illumination effects. In summary, our main contributions are:

- A diffusion inversion method targeting editability through intrinsic-channel conditions;
- Intrinsic-channel disentanglement for streamlined editing;
- Applications to diverse tasks, including appearance editing, object insertion and removal, and relighting of indoor scenes, showing automatic resolution of global illumination effects.

2 Related work

Image generation. Over the past decade, generative models for image synthesis have gained significant attention. Early approaches focused on variational auto-encoders [Kingma 2013], generative adversarial networks [Goodfellow et al. 2014; Karras et al. 2019], and normalizing flows [Kobyzev et al. 2020]. More recently, diffusion models [Ho et al. 2020; Sohl-Dickstein et al. 2015] have emerged as the state of the art, offering unprecedented image quality and

diversity through training scalability [Nichol et al. 2021; Peebles and Xie 2023; Po et al. 2024; Ramesh et al. 2022; Rombach et al. 2022]. A central challenge to image authoring is control over the generation [Bhat et al. 2024; Zhang et al. 2023b].

Generative image editing. Generative methods have quickly gained attention for the decades-old challenge of realistic image editing [Abdal et al. 2019; Avrahami et al. 2022; Collins et al. 2020; Shen et al. 2020; Sheynin et al. 2024; Zhu et al. 2016]. Edits can be specified through various modalities. A common one is text, where the user describes the desired modification to an image through natural language [Brooks et al. 2023; Deutch et al. 2024; Sheynin et al. 2024]. While simple to use, text only supports high-level editing, making it challenging to obtain precise desired edits. Other approaches propose more localized modification, via cut and paste [Alzayer et al. 2024] or dragging [Endo 2022; Mou et al. 2024; Pan et al. 2023b; Pandey et al. 2024], offering a more precise interface.

Diffusion inversion and prompt optimization. DDIM inversion [Song et al. 2021] approximately encodes the original image into the noise that initiates the DDIM diffusion sampling, and is widely used in editing tasks. Null-text inversion (NTI) [Mokady et al. 2023] improves identity preservation by introducing pivot tuning for the null text embeddings, at the cost of more expensive inference-stage optimization. Negative-text inversion [Miyake et al. 2023] and the work of Han et al. [2023] alleviate the computation cost of NTI, with a trade-off in identity preservation. Several works have proposed performance improvements for the noise inversion process [Garibi et al. 2024; Pan et al. 2023a]. Edit-friendly DDPM [Huberman-Spiegelglas et al. 2024] is fast and can achieve accurate image reconstruction. However, it can over-entangle the inverted noise with the image, causing ghosting artifacts during editing (as we show in Fig. 8). In our work, we adopt exact DDIM inversion [Hong et al. 2024]. Although it involves an optimization for each diffusion step, it maps the image to a compact, single initial noise and guarantees identity preservation and good editability. We further improve editability via a prompt-tuning stage [Chung et al. 2023; Gal et al. 2022; Kawar et al. 2023; Mahajan et al. 2024] before inversion. Unlike previous prompt-tuning methods such as Imagic [Kawar et al. 2023], ours encodes a broader range of intrinsic image information through additional supervision, resulting in more controllable edits.

Intrinsic decomposition and re-rendering. Several recent works make progress in intrinsic image decomposition [Barrow et al. 1978], leveraging advances in diffusion models [Chen et al. 2025; Kocsis et al. 2024b; Luo et al. 2024]. Our work builds atop the RGB↔X framework [Zeng et al. 2024a] which uses intrinsic channels (albedo, normals, etc.) to control image generation and editing. However, a straightforward application of this framework suffers from loss of identity and from the need to make aligned edits to several channels at once. We tackle these limitations through noise inversion and prompt-embedding optimization, preserving the identity and naturally blending the edits without modifying the rest of the image.

Single-image relighting. Relighting is a challenging task that requires reasoning about geometry, appearance, and lighting. Existing methods often focus on constrained scenarios, such as the relighting of portraits [Nestmeyer et al. 2020; Ponglertnapakorn et al. 2023; Sun et al. 2019; Zhang et al. 2024c], single objects [Jin et al. 2024; Zeng et al. 2024b], outdoor [Griffiths et al. 2022] and indoor scenes [Li et al. 2022; Murmann et al. 2019; Zhang et al. 2024b]. These methods typically employ an explicit lighting model, encoded by an environment map or some parametric light-source representation. Inspired by classical intrinsic image decomposition [Barrow et al. 1978], recent approaches started employing shading (irradiance) maps as their lighting representation, notably for relighting outdoor scenes [Kocsis et al. 2024a; Yu et al. 2020] and compositing [Zhang et al. 2024a]. Such "shading map" representation offers several advantages. Compared to spherical (HDR) environment maps, shading maps have lower dynamic range and are the same size as the image, simplifying their ingestion into neural networks and enabling concatenation. Our method adopts this shading map—estimated by RGB↔X—as lighting representation and offers a more versatile editing framework than methods explicitly designed for relighting.

Object insertion/removal. Adding or removing content is a staple in the image editing toolbox [Fielding 2013; Niu et al. 2021]. While seamless blending [Burt and Adelson 1983; Farbman et al. 2009; Pérez et al. 2003] and harmonization [Sunkavalli et al. 2010; Tsai et al. 2017; Xue et al. 2012] have been investigated for decades, recent work often fine-tunes diffusion models in a supervised manner [Chen et al. 2023; Liang et al. 2024; Winter et al. 2024; Zhang et al. 2023a], specifying the object to insert/remove through an image and/or mask. Unlike our method, these approaches are specialized to this task. Closest to our work is ZeroComp [Zhang et al. 2024a] which proposes to composite the intrinsic channels of the foreground object and background before generating the edited version.

Material editing. Materials are crucial to a scene's appearance. While their editing is trivial in 3D, it is difficult on a 2D image due to potentially complex global-illumination interactions. Recent work focuses on material editing on objects, either through sliders [Delanoy et al. 2022; Sharma et al. 2024] or image exemplars [Cheng et al. 2025]. Material editing at the scene level has been shown in RGB↔X but suffers from the identity drifts mentioned earlier.

3 Method

Our goal is to leverage the strong natural-image priors of large generative models for realistic image manipulation on a variety of tasks, such as object insertion/removal, material editing, and relighting. We use the intrinsic-image "latent space" of the RGB→X→RGB diffusion pipeline [Zeng et al. 2024a] which (i) decomposes an image into intrinsic channels (albedo, normal, roughness, irradiance) and (ii) recomposes them after edits. However, this pipeline has critical limitations that we must address to make it practicable.

Identity shift. The immediate challenge we face stems from the generative nature of the X→RGB rendering model. Appending that model to the RGB→X decomposition introduces randomness in the middle of the pipeline. In a generative setting, that randomness is necessary as it enables sampling from the entire distribution of images consistent with a given set of intrinsic channels. For our image-editing application that randomness is harmful: it causes identity drifts in the output RGB image—even without edits! To avoid loss of identity, we need to anchor the X→RGB model to

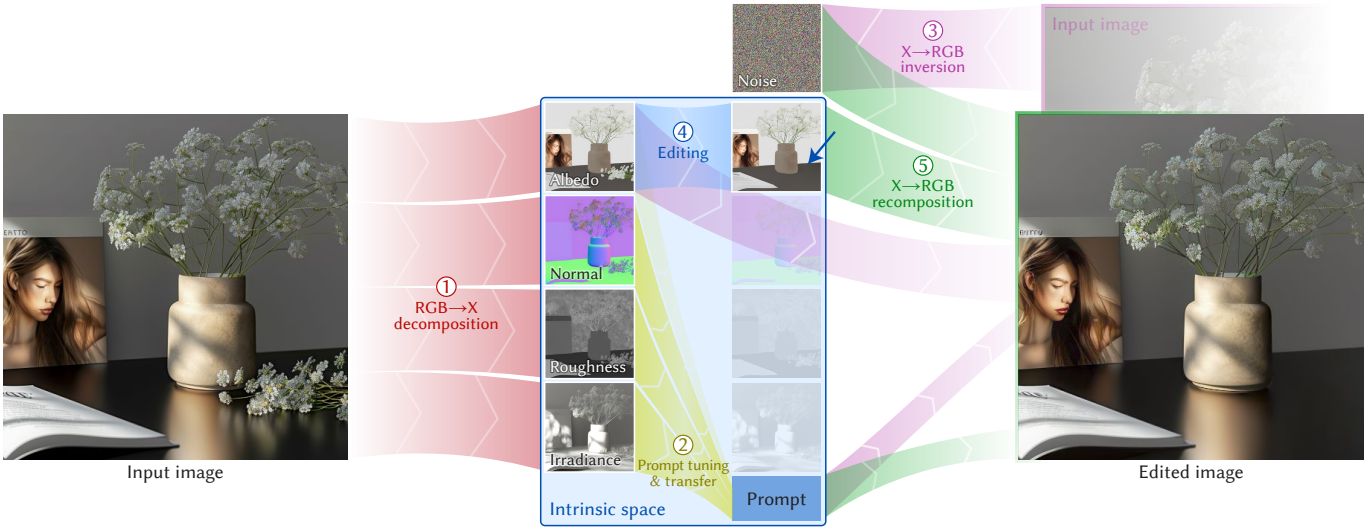

Fig. 3. **IntrinsicEdit overview.** We outline our intrinsic-space editing pipeline, here showing the removal of flowers by manipulating the albedo channel. ① We run RGB→X to decompose the input image into intrinsic channels. ② We tune the prompt embedding to the image and channels (Section 3.2). We also select a subset of channels for editing (here albedo only); any other channels that are entangled with that subset for the desired edit are transferred to the prompt and subsequently dropped (Section 3.3). This step allows us to edit a single channel while preserving information from the rest. ③ We perform exact X→RGB inversion w.r.t. the remaining conditions, i.e. kept channels and optimized prompt. This step finds the noise map that, together with the conditions, accurately reconstructs the input image (Section 3.1). ④ We can now perform the desired edit by manipulating the selected channels. ⑤ Finally, we feed those channels to the X→RGB model, along with the optimized prompt and inverted noise, to synthesize the edited image (Section 3.4). This pipeline allows us to alter only certain image modalities (e.g. material), while automatically propagating the changes to a realistic result and preserving untouched aspects.

reproduce the input image from the initial intrinsic decomposition. This is an inversion problem—computing the starting noise for X→RGB inference that yields a target output image for given (intrinsic) conditions. We perform exact diffusion inversion [Hong et al. 2024] on X→RGB to find that noise (Section 3.1). Once found, we fix it; editing then involves manipulating the intrinsic channels and resynthesizing by running X→RGB using that same noise.

Image/noise entanglement. The ability to accurately reconstruct the input image from the initial intrinsic channels does not necessarily imply that manipulating those channels will produce a plausible edited result. In particular, we observed that the inversion often "bakes" a lot of image-specific information into the noise, which leads to unrealistic editing results containing artifacts or ghost features from the input image. We attribute this to the inverted noise being outside the Gaussian-distribution "comfort zone" of the X→RGB model, (i) as a result of weak or inaccurate conditioning from the channels (e.g. unsupported materials like fabrics), (ii) due to their imprecise RGB→X estimation, or (iii) due the input image being outside both models' distributions. As a remedy, to bring the noise closer to its expected distribution and improve editability, prior to inversion we tune the (originally unused) X→RGB text-embedding conditioning to encode image information that the inversion would otherwise bake into the noise (Section 3.2).

Inter-channel entanglement. Being equipped with the reconstruction noise and necessary conditions, we can start manipulating intrinsic channels to achieve our desired edit. This is where we hit our third problem, which is a general limitation of intrinsic channels: they are partially entangled with one another. For example, removing an object requires careful, aligned editing of *all* channels! While inpainting the albedo channel may be simple, plausibly adjusting the irradiance demands an infeasible light simulation. The blessing of providing dense, semantic conditioning comes with the curse of having to edit all pixels in sync; otherwise, conflicts among channels will lead to artifacts in the X→RGB output (see Fig. 9, bottom row).

Our solution is simple: we drop any channels that conflict with the desired edit on our chosen channel(s). However, inversion w.r.t. a reduced number of conditions can lead to the aforementioned noise-baking problem. We apply a similar remedy: we optimize the prompt to take over the conditioning from the channels that are to be dropped. This optimization effectively transforms the pixel-precise intrinsic conditioning to a more abstract one (Section 3.4). While this solution significantly compresses the amount of conditioning information, it maintains editability and makes the entire pipeline practical by providing freedom to select the most suitable editing modality while delivering plausible results in our experiments.

Overview. Figure 3 illustrates our editing pipeline. After obtaining an RGB→X intrinsic decomposition of the input image, we optimize the X→RGB prompt embedding to (i) tune it to the image and intrinsic conditions and (ii) take over conditioning from edit-entangled channels which are subsequently dropped. We then invert X→RGB w.r.t. the kept channel(s) and optimized prompt, to obtain a reconstruction noise map. The map remains fixed throughout the editing which involves manipulating the kept channel(s) and rendering out a result by invoking X→RGB. Next, we describe these steps in detail.

3.1 X→RGB inversion

X→RGB is a latent diffusion model [Rombach et al. 2022] that reverses an iterative process

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\varepsilon} \tag{1}$$

which gradually corrupts the latent-space representation $\mathbf{z}_0$ of an image with Gaussian noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. The noising schedule $\bar{\alpha}_t \in [0, 1]$ is a function of the time step $t \in [0, T]$, such that $(\bar{\alpha}_0, \bar{\alpha}_T) = (1, 0)$ and thus $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$. The model implements a neural network $\boldsymbol{\varepsilon}_\theta$, with parameters θ, which predicts the noise in a given noisy latent $\mathbf{z}_t$ and which is trained with the objective

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{z}_0,t,\boldsymbol{\varepsilon}}\left\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta\big(\mathbf{z}_t, t, \mathbf{c}_\mathrm{i}, \mathbf{c}_\mathrm{p}\big)\right\|^2. \tag{2}$$

The model is conditioned on a text-prompt embedding $\mathbf{c}_\mathrm{p}$ and a set $\mathbf{c}_\mathrm{i}$ of (latent representations of) intrinsic channels—albedo, normal, roughness, irradiance. We omit metallicity which we found unreliable, as also pointed out by Zeng et al. [2024a]. Note also that they parameterize the model to predict velocity instead of noise [Salimans and Ho 2022].

Given a set of conditions, we can apply deterministic (DDIM) sampling [Song et al. 2021] to iteratively transform an initial noise sample $\mathbf{z}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$ to a clean (latent) image $\mathbf{z}_0$ via the recurrence

$$\mathbf{z}_{t-1} = \sqrt{\alpha_{t-1}\alpha_t^{-1}}\mathbf{z}_t + \left(\sqrt{\alpha_{t-1}^{-1}-1} - \sqrt{\alpha_t^{-1}-1}\right)\cdot\boldsymbol{\varepsilon}_\theta\big(\mathbf{z}_t, t, \mathbf{c}_\mathrm{i}, \mathbf{c}_\mathrm{p}\big). \tag{3}$$

We can think of this T-step sampling process as a neural photorealistic renderer that generates an image given a Gaussian-noise sample, a set of intrinsic channels, and a prompt embedding:

$$\mathbf{z}_0 = \mathrm{X{\to}RGB}\big(\mathbf{z}_T, \mathbf{c}_\mathrm{i}, \mathbf{c}_\mathrm{p}\big). \tag{4}$$

To be able to reconstruct our input image without any edits, we need to invert the above rendering function to obtain the noise $\mathbf{z}_T$ that reproduces the image's clean latent $\mathbf{z}_0$ given conditions $\mathbf{c}_\mathrm{p}$, $\mathbf{c}_\mathrm{i}$. We use exact DDIM inversion [Hong et al. 2024] which we found to work significantly better than faster alternatives such as naive DDIM inversion or edit-friendly DDPM inversion [Huberman-Spiegelglas et al. 2024]. Given $\mathbf{z}_0$, $\mathbf{c}_\mathrm{i}$, and $\mathbf{c}_\mathrm{p}$, the inversion performs gradient-descent optimization of the trajectory of latents $\{\mathbf{z}_t\}_{t=1}^T$ using the following recurrence, starting from $t = 1$:

$$\mathbf{z}_t = \arg\min_{\mathbf{z}'_t}\Big\|\mathbf{z}_{t-1} - \underbrace{\mathbf{z}'_{t-1}(\mathbf{z}_t, t, \mathbf{c}_\mathrm{i}, \mathbf{c}_\mathrm{p})}_{\text{Eq. (3)}}\Big\|^2, \tag{5}$$

taking $\mathbf{z}_{t-1}$ obtained in the previous step. That is, it finds the point $\mathbf{z}_t$ that is mapped to $\mathbf{z}_{t-1}$ by the DDIM sampling in Eq. (3).

3.2 Prompt tuning

Exact inversion allows us to accurately reconstruct the input image but bakes into the optimized noise any image-identity information that is not contained in the conditions. This becomes particularly problematic for editing tasks, where much can simply not be altered because it is "fixed" by that noise. As a result, the X→RGB model can yield artifacts after intrinsic editing. To avoid over-entangling the noise with the image, *prior to inversion* we tune the otherwise unused prompt embedding $\mathbf{c}_\mathrm{p}$ to pick up image-identity features absent from the intrinsic conditions $\mathbf{c}_\mathrm{i}$ using the following loss:

$$\mathcal{L}_\mathrm{tune}(\mathbf{c}_\mathrm{p}) = \mathbb{E}_{t,\boldsymbol{\varepsilon}}\left\|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta\left(\mathbf{z}_t, t, \mathbf{c}_\mathrm{i}, \mathbf{c}_\mathrm{p}\right)\right\|^2. \tag{6}$$

This is the same as the training loss in Eq. (2), this time optimizing for the prompt $\mathbf{c}_\mathrm{p}$ with frozen model parameters θ, given intrinsic conditions $\mathbf{c}_\mathrm{i}$ and input image $\mathbf{z}_0$. At each optimization iteration, we sample a time step t uniformly, and noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$, and compute the latent $\mathbf{z}_t$ using Eq. (1). This prompt tuning pushes $\mathbf{c}_\mathrm{p}$ to contain as much residual information as possible about the input image without over-fitting to a single initial noise z_T.

It is also possible to optimize the estimated intrinsic conditions using the same loss, to align them better with the image and further mitigate feature noise baking. Unfortunately, such naive optimization makes them uninterpretable and thus uneditable.

3.3 Channel-to-prompt transfer

Having a tuned prompt and correspondingly inverted noise allows us to start editing our image though the intrinsic channels. For example, small object-color adjustments are achievable by manipulating only the albedo channel. However, more substantial color edits require carefully updating indirect lighting effects in irradiance, and removing or inserting objects requires non-trivial updates to *all* channels, including shading and shadows in irradiance—a task arguably even less practical than directly manipulating the input image.

We address this problem as follows. For the given editing task, we first identify a subset of channels most suitable for direct manipulation, then drop any other entangled channels that would require manual adjustment. For example, adding/removing objects impacts all channels while albedo editing may not impact normals. Consequently, however, inversion with fewer condition could reintroduce the noise entanglement discussed in Section 3.2. We apply a similar solution: we transfer the information from the dropped channels to the prompt embedding. We achieve this by optimizing the embedding $\mathbf{c}_\mathrm{p}$ such that X→RGB generation with the kept channels and the prompt yields a similar result to generation with all initial channels $\mathbf{c}_\mathrm{i} = \{\mathbf{c}_{\mathrm{i}+}, \mathbf{c}_{\mathrm{i}-}\}$ (kept $\mathbf{c}_{\mathrm{i}+}$ and dropped $\mathbf{c}_{\mathrm{i}-}$) and initial null prompt $\emptyset_\mathrm{p}$. The prompt-optimization loss is thus

$$\begin{aligned}\mathcal{L}_\mathrm{transfer}(\mathbf{c}_\mathrm{p}) = \mathbb{E}_{t,\boldsymbol{\varepsilon}}\Big\|&\boldsymbol{\varepsilon}_\theta\big(z_t, t, \{\mathbf{c}_{\mathrm{i}+}, \mathbf{c}_{\mathrm{i}-}\}, \emptyset_\mathrm{p}\big) - \\ &\boldsymbol{\varepsilon}_\theta\big(z_t, t, \{\mathbf{c}_{\mathrm{i}+}, \emptyset_\mathrm{i}\}, \mathbf{c}_\mathrm{p}\big)\Big\|^2,\end{aligned} \tag{7}$$

where $\emptyset_\mathrm{i}$ means using null intrinsic conditions in place of the dropped channels $\mathbf{c}_{\mathrm{i}-}$. Note that Eq. (7) requires the model $\boldsymbol{\varepsilon}_\theta$ to support any combination $\{\mathbf{c}_{\mathrm{i}+}, \emptyset_\mathrm{i}\}$ of valid and null conditions. Luckily, X→RGB does as it was trained with channel dropout [Zeng et al. 2024a].

Intuitively, this optimization aims to make the prompt have the same effect as using the dropped intrinsic conditions but without requiring their per-pixel editing. The prompt $\mathbf{c}_\mathrm{p}$ now contains an abstract representation of all (dropped) conditions that are to be preserved, while the intrinsics $\mathbf{c}_{\mathrm{i}+}$ explicitly represent conditions to be edited, at the same time minimizing image-specific baking into the inverted noise. This disentangled representation allows us to edit the image in various ways while preserving its original identity.

In practice we perform a single prompt optimization, with a combined loss

$$\mathcal{L}_{\text{prompt}}(\mathbf{c}_\text{p}) = \mathcal{L}_{\text{tune}}(\mathbf{c}_\text{p}) + \lambda\, \mathcal{L}_{\text{transfer}}(\mathbf{c}_\text{p}), \tag{8}$$

where λ balances between tuning and transfer; we use $\lambda \in [0.1, 10]$ in our experiments.

3.4 Intrinsic editing and final synthesis

Given the kept intrinsic conditions $\mathbf{c}_{\text{i+}}$, the optimized prompt $\mathbf{c}_\text{p}$, and the noise $\mathbf{z}_T$ inverted for them, we can finally edit the intrinsics. The set of kept intrinsics varies per application, as we will detail in Section 4 below. Finally, we resynthesize an edited image using the edited intrinsics $\mathbf{c}_{\text{i+}}^{\text{edited}}$:

$$\mathbf{z}_0^{\text{edited}} = \text{X}{\to}\text{RGB}\big(\mathbf{z}_T, \{\mathbf{c}_{\text{i+}}^{\text{edited}}, \emptyset_\text{i}\}, \mathbf{c}_\text{p}\big). \tag{9}$$

We find that this approach leads to good image identity preservation while enabling precise and better disentangled manipulation of different image modalities.

Note that the inversion anchors X→RGB to the input RGB image and its corresponding RGB→X intrinsic decomposition, making the internals of our pipeline deterministic (except for the inherent randomness of stochastic inversion and prompt optimization). The full pipeline remains probabilistic: It takes a (random) noise input at the RGB→X entry point, and the edited result may vary depending on the estimated intrinsic channels as they are propagated to X→RGB.

Diffusion guidance. Guidance is widely used in diffusion models to boost generation quality. For X→RGB inversion, we do not apply guidance. During inference after editing, instead of classifier-free guidance (CFG) [Ho and Salimans 2022], in Eq. (3) we replace $\boldsymbol{\varepsilon}_\theta$ by

$$\boldsymbol{\varepsilon}_\theta^{\text{guided}} = \omega\, \boldsymbol{\varepsilon}_\theta\big(\mathbf{z}_t, t, \mathbf{c}_\text{i}^{\text{edited}}, \mathbf{c}_\text{p}\big) + (1-\omega)\, \boldsymbol{\varepsilon}_\theta\big(\mathbf{z}_t, t, \mathbf{c}_\text{i}, \mathbf{c}_\text{p}\big). \tag{10}$$

This form of guidance is similar to that in negative-prompt inversion [Miyake et al. 2023] but uses the initial intrinsic condition $\mathbf{c}_\text{i}$ instead of a null condition. We use guidance scale $\omega = 1.5$, following Zeng et al. [2024a] who use the same scale for CFG in X→RGB.

4 Results

We now present an evaluation of our approach on four applications: material editing, object removal, object insertion, and relighting. Additionally, we show quantitative evaluations for a subset of applications, as well as ablations to validate our inversion and prompt optimization strategies. Our supplemental document contains an expanded set of results.

Implementation details. We implemented our method atop the public code and models of Zeng et al. [2024a]. We run our prompt optimization with the loss in Eq. (8) for 200 iterations using AdamW optimizer [Loshchilov and Hutter 2017] and learning rate 0.1. To invert the 50-step X→RGB inference w.r.t. the optimized prompt, we follow the backward Euler DDIM solver of Hong et al. [2024], performing 2-3 optimization iterations per diffusion step, seeded by naive DDIM inversion. For roughness editing, we use guidance strength $\omega = 6$ and prompt-loss balance $\lambda \in [1, 10]$ ($\lambda = 1$ in Fig. 4) to ameliorate the X→RGB model's weakness and enforce the edit.

The original RGB→X→RGB pipeline of Zeng et al. [2024a] is a baseline in all our results. For that baseline we use their *inpainting* X→RGB model for all applications except relighting, as we found it to work significantly better than their base X→RGB model in that pipeline, especially for object removal and insertion. The inpainting model takes the input image, a box mask, and the intrinsic channels, and renders an output only inside the mask. We compute the mask by doubling the bounding box of the *edit mask* (difference between original and edited channel(s)) along each dimension, to accommodate for illumination effects; substantial edits can yield masks that cover the entire image. We always use the base (non-inpainting) X→RGB model in our pipeline.

Test data. Most images in our qualitative evaluation are obtained from a stock image database. A small subset is from the Hypersim [Roberts et al. 2021], MIT Indoor [Torralba and Sinha 2009], and InteriorVerse [Zhu et al. 2022] synthetic datasets, and we captured a few images ourselves using smartphone. Note that we did not specifically aim to collect synthetic-looking stock images (known synthetic is top row in Fig. 4, from HyperSim), but did focus on indoor imagery. We do not possess intrinsic-decomposition or edit ground truths for any of the images in our qualitative evaluation.

For real-world quantitative evaluation we use an object-removal dataset of 12 image pairs produced by taking pictures before and after manual object placement. We also evaluate material editing on a dataset derived from 10 synthetic 3D scenes, produced by changing an object's albedo or roughness, and rendering 14 before/after pairs.

Channel organization. For each edit we show, we specify which channels are kept (✔), kept and edited (✦), or dropped (✘) in the inline table below. Any dropped channels are transferred to the prompt as per Section 3.3. For relighting, we found that transferring the unedited irradiance to the prompt improves the inverted-noise disentanglement and the plausibility of edited results. For normal editing, we need to drop the albedo because geometry can change significantly; our supplemental document shows drastic normal edits. For our quantitative evaluation of roughness editing, we drop the normal whose RGB→X estimation is unstable on specular surfaces. We do object insertion via only albedo or via both albedo and normal.

Edit	Albedo	Normal	Roughn.	Irrad.
Color	✦	✔	✔	✘
Normal	✘	✦	✘	✘
Roughness	✔	✔/✘	✦	✘
Relighting	✔	✔	✔	✦
Removal	✦	✘	✘	✘
Insertion	✦	✦/✘	✘	✘

For original RGB→X→RGB we follow the same channel organization, but we do not transfer dropped channels to the prompt.

4.1 Qualitative evaluation

Material editing. Material editing targets the modification of surface color (texture), normal, or roughness. Such editing is greatly simplified in our framework where these intrinsic properties are directly available for manipulation. Figure 4 shows that our method provides fine-grained editing control and generates results that harmonize better with the surrounding environment. We observe that it handles reflection of edited surfaces well (top two rows), and our normal editing correctly infers the right material for the modified kitchen island (third row). Finally, making a shiny floor matte achieves a realistic result that preserves the lighting in the scene.

Fig. 4. **Material editing.** We compare our method against two intrinsic-image methods: original RGB→X→RGB [Zeng et al. 2024a] and intrinsic image diffusion [Kocsis et al. 2024b], and two prompt-based methods: Grounded-Instruct-Pix2Pix [Shagidanov et al. 2024] and TurboEdit [Deutch et al. 2024]. Prompt-based methods fail to provide fine-grained control. Ours is the only one that allows for precise manipulation of individual material properties, preserving identity and harmonizing the edits much better than prior intrinsic-space approaches. Notice the red wall in the top row matching the original material properties while correctly adjusting the color, including the reflection on the counter. The second row shows texture editing on the armchair and addition of two pillows, preserving the lighting and scene identity. In the third row, our approach automatically extends the wooden floor and preserves the kitchen island color despite editing only the normal map. The bottom row shows roughness editing, making the floor more matte and adjusting the reflections.

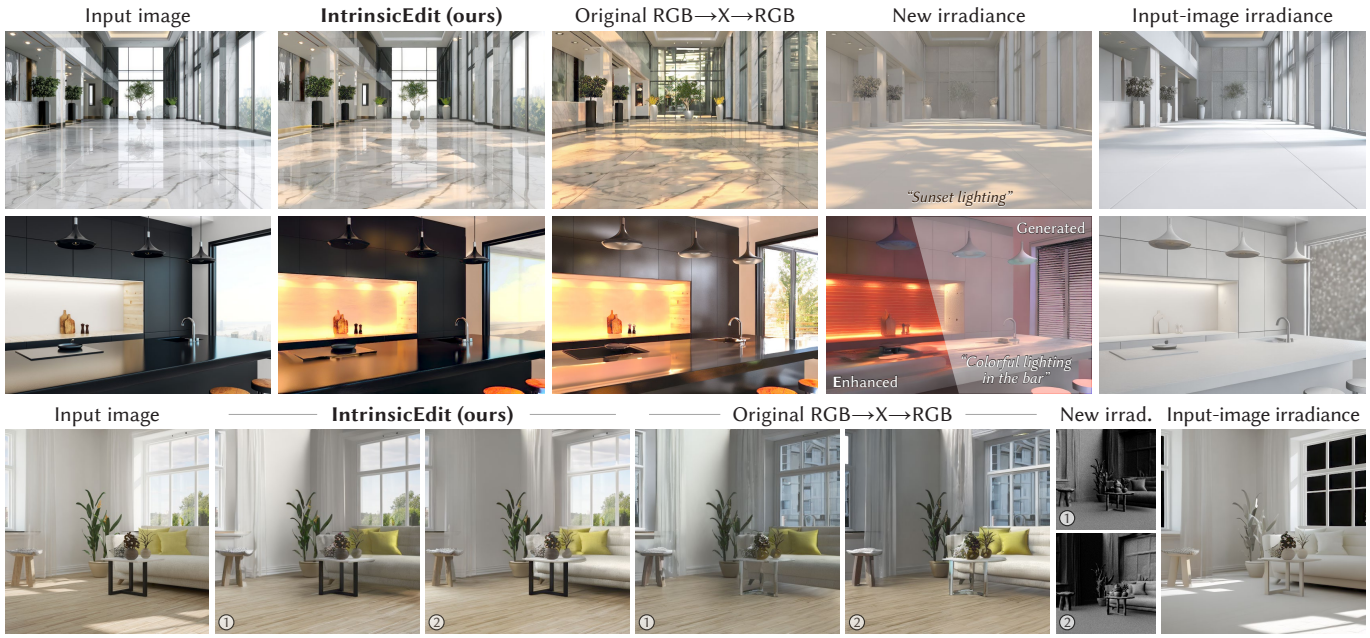

Fig. 5. **Relighting.** In the top two rows we generate a new irradiance channel via prompting as described in Section 4.1. In the bottom row we generate novel irradiance maps using the volumetric shading model of OutCast [Griffiths et al. 2022]. Our method handles the new lighting condition more naturally than original RGB→X→RGB relighting [Zeng et al. 2024a], even when the change is drastic (second row), and better preserves the identity of the scene content.

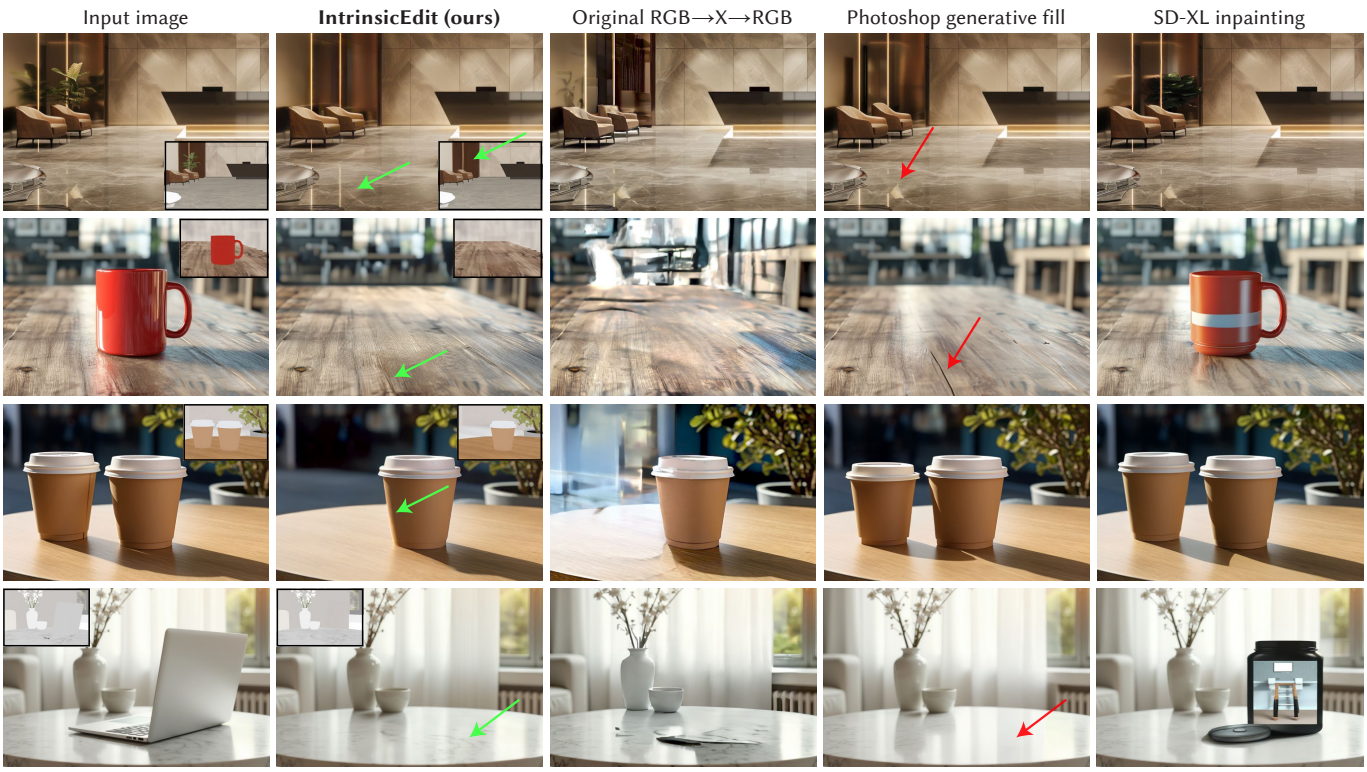

Fig. 6. **Object removal.** We compare against original RGB→X→RGB [Zeng et al. 2024a], Photoshop generative fill [Adobe Inc. 2024], and Stable Diffusion XL inpainting [Stability AI 2023]. Without being specialized for this task, our method performs on par with or better than prior work. In the top row, notice the correct removal of the plant reflection from the marble floor. In the second and fourth rows, the table's texture is preserved better. In the third row, our method successfully removes the left cup, including the shadow it casts on the other cup, while previous methods even struggle to remove the cup completely.

The only other method with results not too far off from ours is the original RGB→X→RGB [Zeng et al. 2024a]. Unfortunately it suffers from all the issues we have addressed in this paper. Zeng et al. had to rely on precise masking to achieve *some* identity preservation, at the cost of compromising the propagation of global illumination effects that the X→RGB model is otherwise able to deliver.

Relighting. Since the irradiance channel can be challenging to modify manually, we use a bootstrapping approach: We sample X→RGB with all intrinsic conditions except irradiance, using text description of the lighting, until we obtain the desired effect. While this approach is already a form of relighting, it leads to identity shifts. To use the new lighting with our method, we simply decompose the obtained image using RGB→X and use the extracted irradiance channel to relight the original input image using our pipeline, after potential stylistic manipulations to the channel.

We show relighting results in Fig. 5 where we change the orientation and color tint of incoming light, e.g. giving an impression of shadows from plants (first row) or turning on kitchen spotlights (second row). We also use the shading model of OutCast [Griffiths et al. 2022] for relighting (third row), where scene depth is estimated, projected into a volumetric model, and queried for novel light directions.

Object removal. We can remove an object from a photograph by inpainting the albedo channel using a photo editor's remove tool. We show results on multiple images in Fig. 6. We can see in all cases that the objects' reflections and shadows are well removed, and that the inpainted regions preserve the background identity better than previous work (second and fourth rows). Image-space inpainting methods have the disadvantage of requiring larger masks that enclose effects like shadows and reflections, which leads to identity shifts in large background regions. Our channel-inpainting masks can tightly bound the object, allowing our method to preserve the surroundings' identity. Note that each Photoshop generative-fill result is the subjectively best one selected from 5 samples.

Object insertion. We perform object insertion by decomposing an object image using RGB→X, or directly extracting its material channels if it is a synthetic object, and pasting them in the target image's channels. Similarly to object removal, we encode the irradiance information in the prompt. We show results in Fig. 7, inserting objects by controlling the albedo map (top row), or both albedo and normal (middle & bottom rows). Our method is the only one that can faithfully harmonize both the object and the rest of the scene, handling reflections and matching the lighting, with little to no compromise in identity.

Fig. 7. **Object insertion.** We compare against original RGB→X→RGB [Zeng et al. 2024a] and existing object-insertion and intrinsic-based methods: IntrinsicComp [Careaga et al. 2023], ZeroComp [Zhang et al. 2024a], Anydoor [Chen et al. 2023], and Poisson cloning [Pérez et al. 2003]. For intrinsic-based methods, we insert the object into the albedo channel (top row) or in both albedo and normal (middle and bottom rows). Despite not being specialized for this task, our approach better harmonizes the inserted object with the rest of the scene. This is particularly visible with the strong directional lighting in the bottom row with the added bust and sphere. In the top row, we duplicate an existing object, observing that our approach successfully handles the table reflection.

4.2 Quantitative evaluation

We evaluate our method quantitatively on synthetic and real-world datasets with ground-truth before/after images. Editing starts from the "before" image and consumes no other data. For each dataset, we present visual comparisons and report PSNR and LPIPS [Zhang et al. 2018] metrics, averaged over the dataset, w.r.t. the "after" ground truth for our method and the same baselines used in Section 4.1.

Synthetic material editing. We evaluate color and roughness editing on renders of synthetic scenes, with dataset sizes of 10 and 4 respectively. Figure 13 summarizes the results. Although not perfect, our method produces images much closer to the reference edited results than all other methods, both visually and numerically.

Real-world object removal. Figure 14 summarizes our results on a real-world object-removal dataset, where we evaluate the results numerically over the whole image and only within the Photoshop generative-fill inpainting mask. Over the whole image, our method is close second behind Photoshop which achieves perfect pixel-value preservation outside its mask; our results suffer from diffusion latent-space encoding/decoding inconsistencies (see Section 4.3 and Fig. 10). Ours performs best within the Photoshop mask which necessarily has to be larger than our albedo-inpainting mask, to include any shadows and reflections. Note that masking can, in principle, be applied to our method, too.

4.3 Ablation studies

Inversion method. Figure 8 demonstrates that exact DDIM inversion [Hong et al. 2024] is crucial to our method's performance and identity preservation (Section 3.1). Replacing the inversion algorithm in our pipeline by edit-friendly DDPM inversion [Huberman-Spiegelglas et al. 2024] preserves its ability to reconstruct the input image, but bakes too much information in its residual noise term, causing severe artifacts. Naive DIM inversion shows good editability but with loss of identity.

Fig. 8. **Inversion method ablation.** Replacing the exact DDIM inversion [Hong et al. 2024] in our pipeline (Fig. 3) with naive DDIM [Song et al. 2021] or edit-friendly DDPM [Huberman-Spiegelglas et al. 2024] inversion has a disastrous effect on its performance, here on object removal.

Prompt optimization. Figure 9 studies the impact of our prompt tuning and transfer optimizations (Sections 3.2 and 3.3). Without tuning or transfer, our method fails to preserve identity and handle illumination after object removal (wrong shadow, background shift). With tuning only, it struggles to estimate lighting (under cabinet) or fails to remove shadows (on remaining cup). With transfer only, it suffers from identity loss (background behind cups). Without inversion, sampling random noise for X→RGB inference, the prompt optimization can still encode part of the identity but cannot reproduce details in the input. Finally, if we keep the unedited channels as explicit conditions instead of transferring them to the prompt, they cause ghosting artifacts after object removal due to the wrong geometry hint. Our prompt transfer allows us to preserve information from entangled channels without having to edit them.

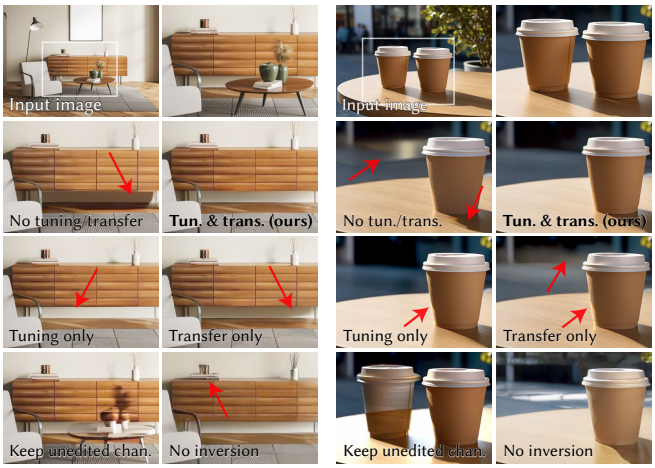

Fig. 9. **Prompt optimization ablation.** Excluding either prompt tuning or channel transfer from our method has negative consequences. Using tuning alone, the method struggles to accurately estimate the correct illumination. On the other hand, using transfer alone compromises identity preservation. Keeping the unedited channels instead of transferring them produces ghosting due to wrong geometry/illumination hints, while skipping inversion and running X→RGB inference with random noise leads to identity loss.

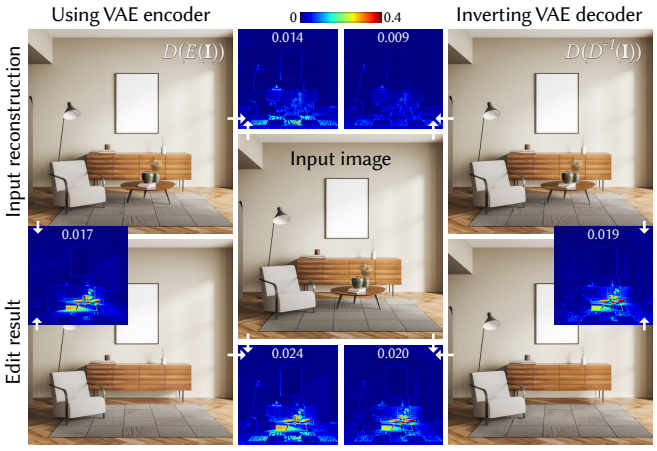

Fig. 10. **Identity preservation.** The variational autoencoder (VAE) of the base foundational (Stable Diffusion) model is a major source of identify shifts in our method. Simply encoding and decoding the input image, $D(E(\mathbf{I}))$, leads to significant loss of fine detail. Inverting the decoder instead, $D(D^{-1}(\mathbf{I}))$, ameliorates the issue but does not eliminate it. Compared to those reconstructions, our edits produce mostly local and predictable differences. These are clear in the L_1 error maps (numbers are image averages).

Fig. 11. **Editing accuracy.** Successful editing requires performing the necessary channel manipulation with accuracy; imprecision can lead to artifacts. Here, going outside the object's boundary with the albedo edit introduces a conflict with the normal-channel condition. Dropping the normal (and roughness) yields a plausible result but with altered object shape.

5 Discussion

Identity preservation. Identity preservation, while significantly improved, is still not perfect. We found that the vast majority of identity shifts are caused by the latent-space encoding of the base Stable Diffusion (SD) 2.1 model [AI 2022] of RGB↔X. We explore the issue in Fig. 10, where we compare two ways to obtain the latent representation of the input image: (1) using the SD encoder and (2) inverting the SD decoder [Hong et al. 2024]. Foregoing any editing, the former option (top left) shows significant loss in high-frequency detail after reconstruction (using the decoder). The latter (top right) yields a more accurate reconstruction, though not perfect—an indication of irrecoverable compression information loss. The error maps comparing our edited results to those reconstructions reveal that the image differences caused by our edits are mostly localized and predictable, accommodating the edit and its effect on the illumination.

Editing accuracy. One of our main goals is to provide pixel-precise editing. However, achieving the expected result requires the user to perform the necessary channel manipulation with accuracy. In Fig. 11 we show the effect of an imprecise albedo manipulation on a color-editing task: editing the albedo beyond the object's boundary introduces a conflict with the geometry condition in the normal channel and yields artifacts. Dropping the normal produces a realistic result but alters the object's shape. Resolving such conflicts and ambiguities is an interesting challenge in generative image editing.

Channel inpainting. Our object-removal pipeline still requires inpainting but on the intrinsic channels (e.g. albedo). In principle, this is a much easier task than inpainting the final image, and future work could investigate inpainting models specific to intrinsic channels to make this step more convenient and robust.

RGB↔X limitations. Although our method shows high-quality results, it inherits some limitations from the RGB↔X models which are trained with limited indoor scene data; further evolution of these models will automatically improve our results. Roughness editing is less reliable than albedo, and metals and transparent objects remain challenging. Another limitation comes from occasionally imperfect RGB→X intrinsic-image decomposition, forcing our inversion process to bake more information into the noise, potentially limiting editing possibilities. More complex materials and light transport effects are another interesting future direction. For example, handling mirrors far from the edited object, multiple reflections, or editing materials such as skin, hair, fur, or fabrics remain challenging. Generally, precise illumination control in images remains an open problem, and is especially difficult for indoor scenes.

While we did not observe quality differences between synthetic- and real-looking images, our model performs best on data closer to the RGB↔X training distribution—indoor scenes; expanding to outdoor scenes and human characters will be key to future improvements as these currently pose a challenge. We show one such

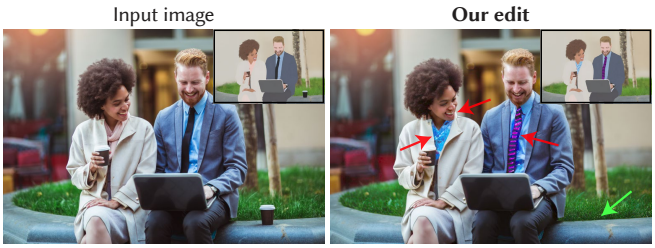

Fig. 12. **RGB↔X limitations.** Images outside the RGB↔X distribution remain challenging. While we can successfully remove the coffee cup, the background is uneditable due to poor intrinsic decomposition. People and garments remain difficult to edit realistically.

out-of-distribution example in Fig. 12. While we can perform some successful editing on that image, the background is uneditable due to poor intrinsic decomposition, and people and their garments are generally not handled well, showing limited editing success.

Speed and resolution. The inference speed of our approach does not match that of feed-forward pipelines. Even though exact inversion is crucial for identity preservation, it can be slow. The resolution of our test images ranges from 512×512 to 1920×1080. For a 512×512 image on an Nvidia H100 GPU, our method takes 75 sec pre-editing (20 sec for 4-channel RGB→X decomposition, 15 sec for 200-step prompt optimization, 40 sec for 50-step inversion) and 5 sec for the 50-step X→RGB inference after editing. Processing a 1920×1080 image takes approximately 500 sec, mostly due to high memory demand for inversion. Large resolutions, such as 4K, are currently infeasible due to memory limitations. A future few-step (distilled) X→RGB model would greatly accelerate inversion as well as prompt optimization (due to smaller range of diffusion steps to sample). We can also reasonably expect future optimizations and hardware improvements to make our method more interactive.

6 Conclusion

Our approach provides pixel-precise generative image editing in intrinsic space. It is made possible by X→RGB diffusion inversion, optimizing both the noise and the prompt embedding to flexibly preserve non-edited information from the input image. We show the capabilities of our framework on a diverse set of non-trivial image manipulation tasks, including object insertion and removal, material editing, and full scene relighting. Despite some remaining limitations, our results demonstrate substantial progress towards unlocking the potential of generative decompose-edit-recompose approaches for image editing.

Acknowledgments

We obtained the 3D scenes for our synthetic material-editing dataset (Fig. 13) from Blenderkit. Tianyu Wang and Qing Liu kindly provided the real-world object-removal dataset (Fig. 14). We thank Zheng Zeng for discussions and help with RGB↔X code and models.

Ethics. We note that realistic generative image manipulation methods like ours could facilitate the spread of misinformation.

References

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2019. Image2stylegan: How to embed images into the stylegan latent space?. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4432–4441.
Adobe Inc. 2024. *Adobe Photoshop*. https://www.adobe.com/products/photoshop.html
Stability AI. 2022. Stable Diffusion 2.1. https://huggingface.co/stabilityai/stable-diffusion-2-1.
Hadi Alzayer, Zhihao Xia, Xuaner Zhang, Eli Shechtman, Jia-Bin Huang, and Michael Gharbi. 2024. Magic Fixup: Streamlining Photo Editing by Watching Dynamic Videos. *arXiv preprint arXiv:2403.13044* (2024).
Omri Avrahami, Dani Lischinski, and Ohad Fried. 2022. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 18208–18218.
Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. 1978. Recovering intrinsic scene characteristics. *Comput. vis. syst* 2, 3-26 (1978), 2.
Shariq Farooq Bhat, Niloy Mitra, and Peter Wonka. 2024. Loosecontrol: Lifting controlnet for generalized depth conditioning. In *ACM SIGGRAPH 2024 Conference Papers*.
Tim Brooks, Aleksander Holynski, and Alexei A Efros. 2023. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18392–18402.
Peter J Burt and Edward H Adelson. 1983. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics* 2, 4 (1983), 217–236.
Chris Careaga, S Mahdi H Miangoleh, and Yağız Aksoy. 2023. Intrinsic Harmonization for Illumination-Aware Compositing Supplementary Material. (2023).
Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. 2023. Anydoor: Zero-shot object-level image customization. *arXiv preprint arXiv:2307.09481* (2023).
Xi Chen, Sida Peng, Dongchen Yang, Yuan Liu, Bowen Pan, Chengfei Lv, and Xiaowei Zhou. 2025. Intrinsicanything: Learning diffusion priors for inverse rendering under unknown illumination. In *European Conference on Computer Vision*. Springer.
Ta-Ying Cheng, Prafull Sharma, Andrew Markham, Niki Trigoni, and Varun Jampani. 2025. Zest: Zero-shot material transfer from a single image. In *European Conference on Computer Vision*. Springer, 370–386.
Hyungjin Chung, Jong Chul Ye, Peyman Milanfar, and Mauricio Delbracio. 2023. Prompt-tuning latent diffusion models for inverse problems. *arXiv preprint arXiv:2310.01110* (2023).
Edo Collins, Raja Bala, Bob Price, and Sabine Susstrunk. 2020. Editing in style: Uncovering the local semantics of gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5771–5780.
Johanna Delanoy, Manuel Lagunas, Diego Gutierrez, and Belen Masia. 2022. A Generative Framework for Image-based Editing of Material Appearance using Perceptual Attributes. *Computer Graphics Forum* 41, 1 (2022), 453–464. doi:10.1111/cgf.14446
Gilad Deutch, Rinon Gal, Daniel Garibi, Or Patashnik, and Daniel Cohen-Or. 2024. Turboedit: Text-based image editing using few-step diffusion models. In *SIGGRAPH Asia 2024 Conference Papers*. 1–12.
Sandra Zhang Ding, Jiafeng Mao, and Kiyoharu Aizawa. 2024. Training-Free Sketch-Guided Diffusion with Latent Optimization. *arXiv preprint arXiv:2409.00313* (2024).
Yuki Endo. 2022. User-controllable latent transformer for StyleGAN image layout editing. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 395–406.
Zeev Farbman, Gil Hoffer, Yaron Lipman, Daniel Cohen-Or, and Dani Lischinski. 2009. Coordinates for instant image cloning. *ACM Trans. Graph.* 28, 3, Article 67 (July 2009), 9 pages. doi:10.1145/1531326.1531373
Raymond Fielding. 2013. *Techniques of special effects of cinematography*. Routledge.
Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618* (2022).
Daniel Garibi, Or Patashnik, Andrey Voynov, Hadar Averbuch-Elor, and Daniel Cohen-Or. 2024. ReNoise: Real Image Inversion Through Iterative Noising. *arXiv preprint arXiv:2403.14602* (2024).
Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
David Griffiths, Tobias Ritschel, and Julien Philip. 2022. OutCast: Outdoor Single-image Relighting with Cast Shadows. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 179–193.
Ligong Han, Song Wen, Qi Chen, Zhixing Zhang, Kunpeng Song, Mengwei Ren, Ruijiang Gao, Anastasis Stathopoulos, Xiaoxiao He, Yuxiao Chen, et al. 2023. Improving Tuning-Free Real Image Editing with Proximal Guidance. *arXiv preprint arXiv:2306.05414* (2023).
Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598* (2022).
Seongmin Hong, Kyeonghyun Lee, Suh Yoon Jeon, Hyewon Bae, and Se Young Chun. 2024. On Exact Inversion of DPM-Solvers. In *Proceedings of the IEEE/CVF Conference*

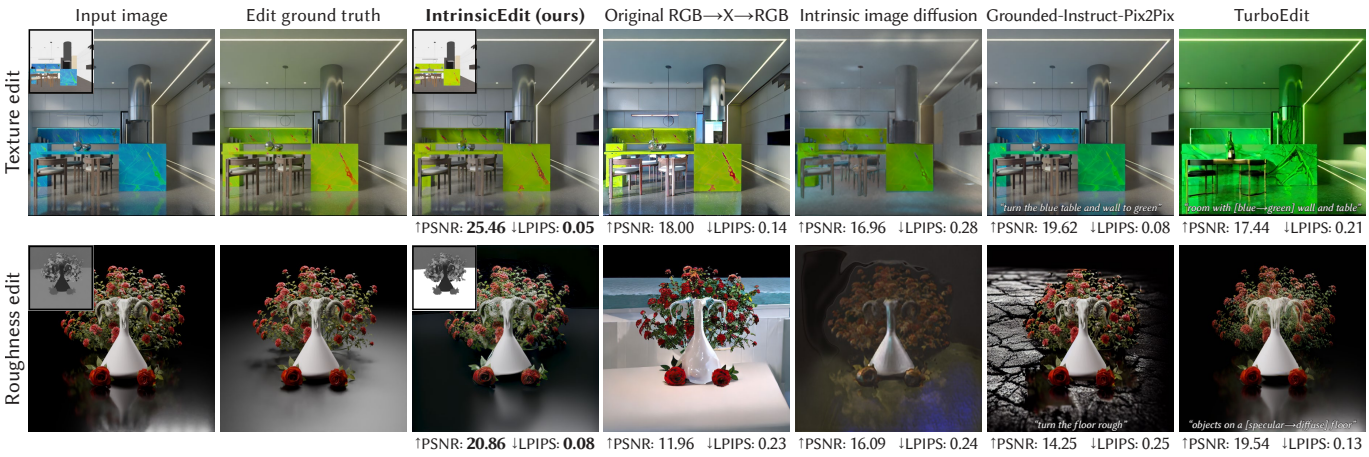

Fig. 13. **Material editing on synthetic images.** On a set of synthetic-scene renders, we compare our method against the same baselines as in Fig. 4, this time also quantitatively w.r.t. ground-truth edit results. The results of our method match the ground truths most closely, both numerically and visually, although not perfectly. The reported metrics are averaged over a dataset of 10 before/after pairs for texture editing and 4 pairs for roughness editing. We stress that we do not use any ground-truth intrinsic channels: editing starts from the "before" image and consumes no other data.

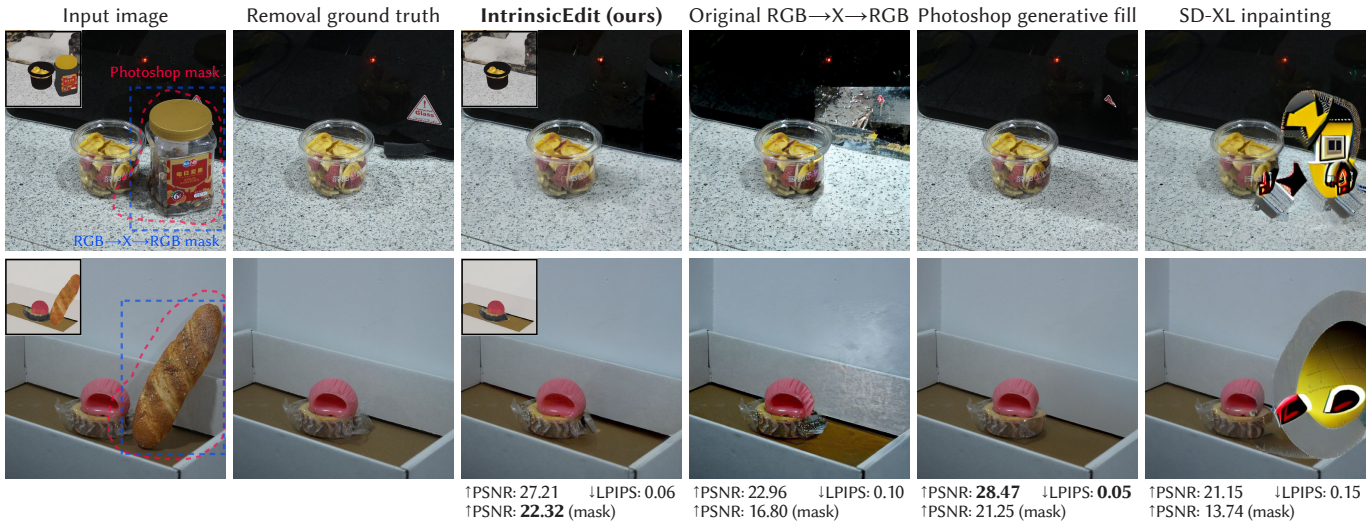

Fig. 14. **Real-world object removal.** We compare against the same baselines as in Fig. 6, this time also quantitatively w.r.t. ground-truth edit results. Photoshop generative fill is the only method that rivals ours, performing best over the entire image thanks to perfect pixel preservation outside its inpainting mask. Inside that mask, our method preserves identity best, textures in particular. The reported metrics are averaged over the dataset of 12 before/after pairs.

on Computer Vision and Pattern Recognition. 7069–7078.

Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. 2024. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12469–12478.

Haian Jin, Yuan Li, Fujun Luan, Yuanbo Xiangli, Sai Bi, Kai Zhang, Zexiang Xu, Jin Sun, and Noah Snavely. 2024. Neural Gaffer: Relighting Any Object via Diffusion. *arXiv preprint arXiv:2406.07520* (2024).

Tero Karras, Samuli Laine, and Timo Aila. 2019. A Style-Based Generator Architecture for Generative Adversarial Networks. In *CVPR*.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. 2023. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6007–6017.

Diederik P Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. 2020. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence* 43, 11 (2020), 3964–3979.

Peter Kocsis, Julien Philip, Kalyan Sunkavalli, Matthias Nießner, and Yannick Hold-Geoffroy. 2024a. Lightit: Illumination modeling and control for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9359–9369.

Peter Kocsis, Vincent Sitzmann, and Matthias Nießner. 2024b. Intrinsic Image Diffusion for Indoor Single-view Material Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5198–5208.

Seonho Lee, Jiho Choi, Seohyun Lim, Jiwook Kim, and Hyunjung Shim. 2024. Scribble-Guided Diffusion for Training-free Text-to-Image Generation. *arXiv preprint arXiv:2409.08026* (2024).

Zhengqin Li, Jia Shi, Sai Bi, Rui Zhu, Kalyan Sunkavalli, Miloš Hašan, Zexiang Xu, Ravi Ramamoorthi, and Manmohan Chandraker. 2022. Physically-based editing of indoor scene lighting from a single image. In *European Conference on Computer*

Vision. Springer, 555–572.
Ruofan Liang, Zan Gojcic, Merlin Nimier-David, David Acuna, Nandita Vijaykumar, Sanja Fidler, and Zian Wang. 2024. Photorealistic Object Insertion with Diffusion-Guided Inverse Rendering. In *ECCV*.
Ilya Loshchilov and Frank Hutter. 2017. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*. https://api.semanticscholar.org/CorpusID:53592270
Jundan Luo, Duygu Ceylan, Jae Shin Yoon, Nanxuan Zhao, Julien Philip, Anna Frühstück, Wenbin Li, Christian Richardt, and Tuanfeng Wang. 2024. Intrinsicdiffusion: joint intrinsic layers from latent diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.
Shweta Mahajan, Tanzila Rahman, Kwang Moo Yi, and Leonid Sigal. 2024. Prompting Hard or Hardly Prompting: Prompt Inversion for Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6808–6817.
Daiki Miyake, Akihiro Iohara, Yu Saito, and Toshiyuki Tanaka. 2023. Negative-prompt inversion: Fast image inversion for editing with text-guided diffusion models. *arXiv preprint arXiv:2305.16807* (2023).
Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. 2023. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6038–6047.
Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. 2024. Dragondiffusion: Enabling drag-style manipulation on diffusion models. In *ICLR*.
Lukas Murmann, Michael Gharbi, Miika Aittala, and Fredo Durand. 2019. A dataset of multi-illumination images in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4080–4089.
Thomas Nestmeyer, Jean-François Lalonde, Iain Matthews, and Andreas Lehrmann. 2020. Learning physics-guided face relighting under directional light. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5124–5133.
Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2021. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).
Li Niu, Wenyan Cong, Liu Liu, Yan Hong, Bo Zhang, Jing Liang, and Liqing Zhang. 2021. Making images real again: A comprehensive survey on deep image composition. *arXiv preprint arXiv:2106.14490* (2021).
Xingang Pan, Ayush Tewari, Thomas Leimkühler, Lingjie Liu, Abhimitra Meka, and Christian Theobalt. 2023b. Drag your GAN: Interactive point-based manipulation on the generative image manifold. In *ACM SIGGRAPH 2023 conference proceedings*.
Zhihong Pan, Riccardo Gherardi, Xiufeng Xie, and Stephen Huang. 2023a. Effective real image editing with accelerated iterative diffusion inversion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 15912–15921.
Karran Pandey, Paul Guerrero, Metheus Gadelha, Yannick Hold-Geoffroy, Karan Singh, and Niloy J. Mitra. 2024. Diffusion Handles: Enabling 3D Edits for Diffusion Models by Lifting Activations to 3D. *CVPR* (2024).
William Peebles and Saining Xie. 2023. Scalable Diffusion Models with Transformers. In *ICCV*.
Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*. 313–318.
Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T Barron, Amit Bermano, Eric Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. 2024. State of the art on diffusion models for visual computing. In *Computer Graphics Forum*, Vol. 43. e15063.
Puntawat Ponglertnapakorn, Nontawat Tritrong, and Supasorn Suwajanakorn. 2023. DiFaReli: Diffusion face relighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 22646–22657.
Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical Text-Conditional Image Generation with CLIP Latents. arXiv:2204.06125 [cs.CV] https://arxiv.org/abs/2204.06125
Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. 2021. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*. 10912–10922.
Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
Tim Salimans and Jonathan Ho. 2022. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations*. https://openreview.net/forum?id=TIdIXIpzhoI
Artur Shagidanov, Hayk Poghosyan, Xinyu Gong, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. 2024. Grounded-Instruct-Pix2Pix: Improving Instruction Based Image Editing with Automatic Target Grounding. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6585–6589. doi:10.1109/ICASSP48485.2024.10446377
Prafull Sharma, Varun Jampani, Yuanzhen Li, Xuhui Jia, Dmitry Lagun, Fredo Durand, Bill Freeman, and Mark Matthews. 2024. Alchemist: Parametric control of material properties with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24130–24141.
Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9243–9252.
Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. 2024. Emu edit: Precise image editing via recognition and generation tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8871–8879.
Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. 2024. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8839–8849.
Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*. PMLR, 2256–2265.
Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.
Stability AI. 2023. Stable Diffusion XL Inpainting. https://huggingface.co/diffusers/stable-diffusion-xl-1.0-inpainting-0.1.
Tiancheng Sun, Jonathan T Barron, Yun-Ta Tsai, Zexiang Xu, Xueming Yu, Graham Fyffe, Christoph Rhemann, Jay Busch, Paul Debevec, and Ravi Ramamoorthi. 2019. Single image portrait relighting. *ACM Transactions on Graphics* 38, 4 (2019), 1–12.
Kalyan Sunkavalli, Micah K Johnson, Wojciech Matusik, and Hanspeter Pfister. 2010. Multi-scale image harmonization. *ACM Transactions on Graphics* 29, 4 (2010), 1–10.
Antonio Torralba and Pawan Sinha. 2009. Recognizing Indoor Scenes. (2009).
Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. 2017. Deep image harmonization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3789–3797.
Daniel Winter, Matan Cohen, Shlomi Fruchter, Yael Pritch, Alex Rav-Acha, and Yedid Hoshen. 2024. ObjectDrop: Bootstrapping Counterfactuals for Photorealistic Object Removal and Insertion. *arXiv preprint arXiv:2403.18818* (2024).
Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. 2025. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision*. Springer, 331–348.
Su Xue, Aseem Agarwala, Julie Dorsey, and Holly Rushmeier. 2012. Understanding and improving the realism of image composites. *ACM Transactions on graphics* 31, 4 (2012), 1–10.
Ye Yu, Abhimitra Meka, Mohamed Elgharib, Hans-Peter Seidel, Christian Theobalt, and William AP Smith. 2020. Self-supervised outdoor scene relighting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*. Springer, 84–101.
Chong Zeng, Yue Dong, Pieter Peers, Youkang Kong, Hongzhi Wu, and Xin Tong. 2024b. Dilightnet: Fine-grained lighting control for diffusion-based image generation. In *ACM SIGGRAPH 2024 Conference Papers*. 1–12.
Zheng Zeng, Valentin Deschaintre, Iliyan Georgiev, Yannick Hold-Geoffroy, Yiwei Hu, Fujun Luan, Ling-Qi Yan, and Miloš Hašan. 2024a. Rgb↔ x: Image decomposition and synthesis using material-and lighting-aware diffusion models. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.
Bo Zhang, Yuxuan Duan, Jun Lan, Yan Hong, Huijia Zhu, Weiqiang Wang, and Li Niu. 2023a. Controlcom: Controllable image composition using diffusion model. *arXiv preprint arXiv:2308.10040* (2023).
Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023b. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3836–3847.
Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2024c. IC-Light GitHub Page.
Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
Xiao Zhang, William Gao, Seemandhar Jain, Michael Maire, David Forsyth, and Anand Bhattad. 2024b. Latent Intrinsics Emerge from Training to Relight. In *NeurIPS*.
Zitian Zhang, Frédéric Fortier-Chouinard, Mathieu Garon, Anand Bhattad, and Jean-François Lalonde. 2024a. Zerocomp: Zero-shot object compositing from image intrinsics via diffusion. *arXiv preprint arXiv:2410.08168* (2024).
Jingsen Zhu, Fujun Luan, Yuchi Huo, Zihao Lin, Zhihua Zhong, Dianbing Xi, Rui Wang, Hujun Bao, Jiaxiang Zheng, and Rui Tang. 2022. Learning-based inverse rendering of complex indoor scenes with differentiable monte carlo raytracing. In *SIGGRAPH Asia 2022 Conference Papers*. 1–8.
Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In *Computer vision–ECCV 2016: 14th European conference, amsterdam, the netherlands, October 11-14, 2016, proceedings, part v 14*. Springer, 597–613.
Shenhao Zhu, Junming Leo Chen, Zuozhuo Dai, Zilong Dong, Yinghui Xu, Xun Cao, Yao Yao, Hao Zhu, and Siyu Zhu. 2025. Champ: Controllable and consistent human image animation with 3d parametric guidance. In *ECCV*. Springer, 145–162.