Jump Restore Light Transport

ANONYMOUS AUTHOR(S)
SUBMISSION ID: 1176

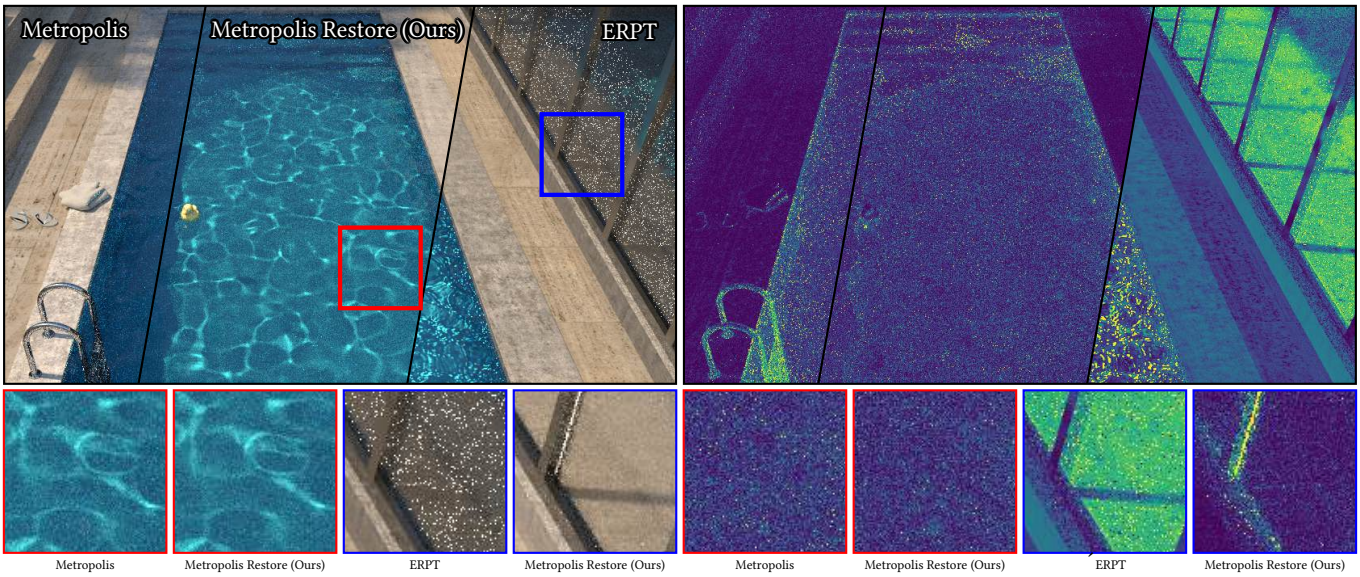

Fig. 1. We present a novel continuous-time Markov chain Monte Carlo (MCMC) framework that *adjusts* an arbitrary family of Markov processes — used solely for *local exploration* — into a global process, which is invariant with respect to a given target distribution. Crucially, our approach allows for the seamless integration of *any* existing MCMC sampler for local exploration. The resulting integrated algorithm consistently outperforms the original method, offering shorter running time, lower error, and reduced variance. In the figure, we depicted an equal rendering time comparison (30 s) of Multiplexed Primary Sample Space Metropolis Light Transport (PSSMLT) [Hachisuka et al. 2014] (left), its integration into our framework (middle), and Energy Redistribution Path Tracing (ERPT) [Cline et al. 2005] (right) for the Swimming Pool scene provided by [Rioux-Lavoie et al. 2020].

Markov chain Monte Carlo (MCMC) algorithms are indispensable when sampling from a complex, high-dimensional distribution by a conventional method is intractable. Even though MCMC is a powerful tool, it is also hard to control and tune in practice. Simultaneously achieving both rapid *local exploration* of the state space and efficient *global discovery* of the target distribution is a challenging task.

In this work, we introduce a novel continuous-time MCMC formulation to the computer science community. Generalizing existing work from the statistics community, we propose a novel framework for *adjusting* an arbitrary family of Markov processes - used for local exploration of the state space only - to an overall process which is invariant with respect to a target distribution.

To demonstrate the potential of our framework, we focus on a simple, but yet insightful, application in light transport simulation. As a by-product, we introduce continuous-time MCMC sampling to the computer graphics community. We show how any existing MCMC-based light transport algorithm can be seamlessly integrated into our framework. We prove empirically and theoretically that the integrated version is superior to the ordinary algorithm. In fact, our approach will convert any existing algorithm into a highly *parallelizable* variant with shorter running time, smaller error and less variance.

CCS Concepts: • **Mathematics of computing** → **Probability and statistics**; **Probabilistic algorithms**; **Stochastic processes**; **Markov processes**; • **Computing methodologies** → **Computer graphics**; **Rendering**; **Ray tracing**; Concurrent algorithms.

Additional Key Words and Phrases: diffusion processes, jump-type Markov processes, light transport simulation

ACM Reference Format:
Anonymous Author(s). 2025. Jump Restore Light Transport. 1, 1 (May 2025), 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM XXXX-XXXX/2025/5-ART
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In light transport simulation, the computation of high-dimensional integrals is essential and is typically performed using Monte Carlo (MC) integration. Traditionally, this method involves generating independent samples within each pixel of the image space. However, a major drawback of this approach is that samples are generated regardless of their actual contribution to the final estimate. Even if a sample has no impact on the result, it is still drawn from a predefined importance distribution, without explicitly considering the value of the target density at the sampled location beforehand.

Markov chain Monte Carlo (MCMC) methods offer a way to address this inefficiency. By constructing a Markov process, sample generation can be guided to better align with the target distribution, allowing for a more structured exploration of the underlying space.

Veach [1997] introduced the Metropolis-Hastings (MH) algorithm, arguably the most popular and widely applicable MCMC method, to the graphics community. Building on this pioneering work, numerous MH-based light transport algorithms have been proposed since then. In fact, every MCMC-based light transport algorithm is actually a MH-variant.

Traditional approaches and their limitations. In its general formulation, MH is a recipe for constructing a Markov chain that is invariant with respect to a desired target distribution π. The key ingredient, which the user can choose (within mild constraints), is the *proposal kernel* ζ. Readers unfamiliar with MH can find a detailed explanation in section 4. For now, it is only important to understand that MH internally simulates a Markov chain Y with transition kernel ζ. This Markov chain, uniquely determined by ζ, is typically chosen so as to explore the state space as rapidly and targeted as possible. Once ζ is for our specific application, it is essentially Y that we would ideally like to use for exploration.

The problem, however, is that ζ, and thus Y, is generally not already π-invariant and therefore is not eventually distributed according to π. MH can be viewed as a procedure that *adjusts* Y so that the resulting chain becomes π-invariant. This is achieved by not blindly following each proposed state transition of Y , but instead accepting or rejecting them based on an *acceptance probability* that depends on π. Details can be found in section 4. The resulting MH chain is π-invariant.

MH is surprisingly easy to implement and often performs quite well in practice. However, there are serious issues that — among other reasons — still prevent its use in production rendering today.

Modern sampling problems, whether in generative AI or rendering, are shaped by two key factors that determine the overall efficiency : *local exploration* and *global discovery*. That is, we want algorithms that explore locally in a rapid and targeted way, while also ensuring global discovery of the target distribution.

MH variants struggle with both goals. On the one hand, the MH chain is not only π-invariant, but even π-reversible [Çinlar 2011; Ethier and Kurtz 2009]. While this is useful for theoretical analysis (e.g., due to favorable spectral properties), it leads in practice to significantly reduced convergence speed [Bierkens 2015]. Reversibility causes excessive backtracking — the MH chain frequently revisits regions it has already explored. Importantly, even if the original Markov chain Y induced by the proposal kernel is nonreversible, this nonreversibility is destroyed by the MH adjustment.

In the practically most relevant Euclidean state spaces, desired (local) exploration is typically modeled via diffusion processes, as they are particularly well-suited to describe particle motion in space. To be used within MH, these must be time-discretized in a way that renders the resulting process a Markov chain from which the proposal kernel ζ can be extracted. The most prominent methods — which we later include in our numerical study in section 9 — have emerged in this way.

Ensuring global discovery is even more severe. If π has separated modes, the MH chain may get trapped in one of them, since the MH adjustment prevents Y from escaping once inside.

Even outside such worst-case settings, effective global discovery requires local exploration to be relocated to new regions of the state space after some time. One might attempt to circumvent this issue by running multiple independent MH chains in parallel. But this approach has its own limitations. All chains could, in theory, get stuck in some mode. Moreover, MH suffers from *start-up bias*: depending on the initial state, it may take time for Y to reach the target distribution. Consequentially, early states of the MH chain must be discarded. As a result, even if many chains are launched in parallel, we might quickly accumulate the desired number of samples, but — in the worst case — none of them is truly representative of the target distribution. This issue can only be resolved through an initialization phase designed to identify suitable initial states.

Instead of relying solely on multiple parallel MH chains, the graphics community often resorts to artificial means of addressing global discovery — for instance, by replacing the "local" proposal kernel ζ with a mixture proposal kernel that includes both small-scale and large-scale moves. We discuss this further in section 5.

Our novel framework and its solution to traditional limitations. To address all of these issues, we propose a framework that subsumes and significantly generalizes MH. Our approach builds on the recently introduced *Restore* algorithm by Wang et al. [2021] in the statistics community. To meet the specific requirements of our domain, we significantly extend the original framework and relax its assumptions — both on the process used for local exploration and on the target distribution. In particular, it is only through this generalization that the use of this method becomes theoretically justified for light transport simulation.

In detail, we allow the use of an arbitrary family of *continuous-time* Markov processes Y^i for local exploration. Global discovery is ensured via a novel transfer mechanism that, after a duration depending on both the local target density and elapsed time, relocates the exploration process to another region of the state space. The starting point of the new local exploration may depend on the *exit point* of the previous one.

The resulting overall process is invariant with respect to the desired target distribution. In this sense, our framework can be seen as an adjustment procedure that turns an arbitrary family of Markov processes into an overall process that is invariant with respect to a given target distribution.

We highlight the following advantages of our framework:

(1) Local exploration via *arbitrary* Markov processes.
(2) Potential nonreversibility of the Markov processes is *preserved*, not destroyed.
(3) In MH, a large step proposal into a high-density region (e.g., a bright region) is likely to be accepted. However, this interrupts the ongoing exploration in the previous region, leading to a bias toward oversampling the new region. In contrast, our approach never terminates local exploration based on global criteria. Instead, we end it based solely on *local conditions*: the target density in the current region and the elapsed exploration time. This allows for better

balance, avoids premature focus on oversampled regions, and maintains continuity in the exploration process.

(4) If the transfer between local explorations does not depend on the exit points of their respective predecessors, all local explorations can be executed in parallel without introducing startup bias into the estimator used in this context.

Outline of this work. In section 2, we provide an overview of related work, with a particular focus on existing MH-based light transport algorithms and the general-purpose MH variants originally introduced in the statistics community that they are based on. In section 3, we briefly review the fundamental principles of MCMC, for completeness. In section 4, we describe the Metropolis–Hastings algorithm. Understanding its internal mechanism is crucial for the comparisons we draw in the numerical study. In section 5, we introduce the global discovery problem in detail and provide an illustrative example. We also describe the artificial workaround that is still used in current MH-based light transport algorithms. In section 6, we present our generalization of the *Restore* framework. We first provide an abstract formulation that holds promise for future methods, and then concretize it into a practically implementable algorithm that can be directly compared to existing techniques. In section 8, we describe the practical setup underlying our numerical study and show how any existing MCMC-based light transport algorithm can be transformed — by integrating it into our framework — into a highly parallelizable variant with shorter runtime, lower error, and reduced variance. In section 9, we finally present our numerical study and provide empirical results that demonstrate the superiority of our framework over traditional methods.

2 RELATED WORK

Seminal work. After MH [Metropolis et al. 1953; Hastings 1970] was introduced to the graphics community in Veach [1997] and made practical by replacing the path space formulation with the (Euclidean) primary sample space in Kelemen et al. [2002], a wide range of MH variants have been proposed.

Diffusion-based proposals. Some are specific choices of proposal kernels that had already been introduced in the statistics community, including discretizations of Langevin [Roberts and Tweedie 1996] and Hamiltonian dynamics [Duane et al. 1987], which led to light transport algorithms such as Luan et al. [2020a] and Li et al. [2015].

MH variants. Others adapt established MH variants to the light transport setting, including charted [Marinari and Parisi 1992; Pantaleoni 2017], delayed rejection [Mira 2011; Rioux-Lavoie et al. 2020], multiple-try [Liu et al. 2000; Segovia et al. 2007], and reversible jump [Green 1995; Bitterli et al. 2017] MH.

bidirectional path tracing (BDPT). BDPT was incorporated into MH by Hachisuka et al. [2014], which formally only modifies the target distribution and allows for strategy-dependent proposal kernels. Most MH variants proposed since then adopt this multiplexed formulation.

The basic Restore framework. Finally, the basic Restore framework that we generalize was originally proposed in Wang [2020]; Wang et al. [2021] and later extended in McKimm et al. [2024].

3 MARKOV CHAIN MONTE CARLO

Given a finite measure π, MCMC is a technique for estimating the integral

$$I := \int f \, \mathrm{d}\pi \tag{1}$$

of a π-integrable function f. More precisely, it is a recipe for constructing an ergodic Markov process with invariant distribution π.

Markov process. A process is a state system evolving over time. In this work, the time domain T will either be discrete, $T = \mathbb{N}_0$, or continuous, $T = [0, \infty)$. Informally, the process is said to be Markov, if at any fixed point in time, the evolution of the process does only depend on the present state, but not on the past.

Invariance. π being an invariant distribution of a Markov process $(X_t)_{t \in T}$ is equivalent to enforcing that once $(X_t)_{t \in T}$ is distributed according to π at a certain time point $s \in T$, every state X_t at a future time point $t \in I \cap (s, \infty]$ will be distributed according to π as well. That is, the distribution of a state is stationary in time after it once coincided with π.

Ergodicity. The ergodicity, on the other hand, will ensure that the long time average of an observation is effectively equal to space averaging with respect to the invariant distribution. That is, given that the invariant distribution π actually exists, ergodicity is equivalent to enforcing that if X_0 is distributed according to π, then

$$A_t := \frac{1}{t} \begin{cases} \sum_{s=0}^{t-1} f(X_s) & , \text{if } T = \mathbb{N}_0 \\ \int_0^t f(X_s) \, \mathrm{d}s & , \text{if } T = [0, \infty) \end{cases} \xrightarrow{t \to \infty} I \tag{2}$$

almost surely for all $f \in \mathcal{L}^1(\pi)$.

This characterization of ergodicity is known as *Birkhoff's ergodic theorem* [Kallenberg 2021, Theorem 25.6]. In light of (2), it is evident why $(A_t : t \in T \setminus \{0\})$ is usually called the *ergodic average estimator* of I. In this work, we will always assume that the processes under consideration exhibit this form of ergodicity. For a technical conditional ensuring that we refer to Meyn and Tweedie [1993].

4 METROPOLIS-HASTINGS ALGORITHM

The Metropolis-Hastings (MH) algorithm is arguably the most popular and widely applicable MCMC method. It is an algorithmic construction of a Markov chain M with invariant distribution π. The procedure of simulating this chain up to a given time $t \in \mathbb{N}_0$ is summarized in Algorithm 4.1.

Algorithmic description. The user has to specify a *proposal kernel* Q. For every state x, $Q(x, \cdot)$ is a probability measure. Now, at each discrete time step, the algorithm is *proposing* a state transition candidate y drawn from $Q(x, \cdot)$, where x is the current state of the chain generated so far. With probability $\alpha(x, y)$, where α is an *acceptance function*, the *proposal* y is *accepted* (line 4) and the current state is set to y. With the opposite probability, $1 - \alpha(x, y)$, the proposal is *rejected* (line 5) and the current state will not be changed (cf. line 7).

Algorithm 4.1 Metropolis-Hastings algorithm with proposal kernel Q and target distribution π.

Input: Initial state x_0 and sample count $t \in \mathbb{N}$.
Output: Realization $(x_0, \ldots, x_{t-1})$ of the MH chain M

```
procedure MetropolisHastingsUpdate(x)
    Sample y from Q(x, ·);                    ← generate the proposal
    Sample u from U_[0,1);                    ← uniform distribution on [0, 1)
    if (u < α(x, y)) return y;                ← with prob. α(x, y) return proposal
    return x;                                 ← with prob. 1 − α(x, y) reject proposal
for (s = 1; s < t; ++s)
    x_s = MetropolisHastingsUpdate(x_{s−1});
```

Acceptance function. The acceptance function is defined as

$$\alpha(x, y) := \min\left(1, \frac{p(y)q(y,x)}{p(x)q(x,y)}\right) \tag{3}$$

where we assume that both the target distribution π and the proposal kernel Q admit densities with respect to a common reference measure λ; that is,

$$\pi(B) = \frac{1}{p_\lambda} \int_B p \, \mathrm{d}\lambda \tag{4}$$

for some density p and a normalization constant $p_\lambda \in (0, \infty)$, and analogously,

$$Q(x, B) = \int_B q(x, \cdot) \, \mathrm{d}\lambda \tag{5}$$

for some density q.

5 GLOBAL DISCOVERY

Local MCMC algorithms - that is, algorithms whose state transitions are confined to local neighborhoods of the current state - often struggle to explore multimodal target distributions. This class of algorithms includes all MH variants whose proposal kernels propose only *small-scale* moves, i.e., transitions limited to the local neighborhood of the current state.

This is a practical problem, since the most common proposal kernels for the MH algorithm are based on time-discretized diffusion processes. They are excellent for local exploration. However, without incorporation of *large-scale* moves, exploration of the whole state space will be slow and could even get stuck in local modes of the target distribution. This is particularly intuitive in the Langevin algorithm - whether Metropolis-adjusted or not - as it is effectively a stochastically perturbed gradient descent update scheme.

Mixture proposal. Especially in light transport simulation, the traditional attempt to address the aforementioned issue and thereby ensure global exploration is to mix in large-scale proposals. For this reason, a typical proposal kernel Q is constructed as a *mixture*:

$$Q_\ell(x, \cdot) := \underset{\text{global}}{\ell\mu} + \underset{\text{local}}{(1-\ell)\zeta}(x, \cdot) \tag{6}$$

Intuitively, ℓ, μ, and ζ are referred to as the *large step probability*, *large step distribution*, and *small step kernel* of the proposal scheme, respectively. By construction, $\ell \in [0, 1]$ controls the probability of performing a *large step* drawn from μ. In order for this proposal to be effective, μ should be capable of making large jumps across the state space, while ζ should focus on thorough local exploration.

Practical limitations. Even though this approach works quite well in practice, it is difficult to prescribe a universally effective choice for the large step probability ℓ. Moreover, according to (6), a *large step proposal* introduces a large-scale move *uninformatively* — that is, without taking into account how productive the current phase of exploration is. While such proposals can be rejected if the target density at the proposed state is significantly lower than at the current state, they may nonetheless divert the exploration away from regions where sampling was proceeding efficiently. Additionally, upon rejection, the chain remains at the previous state, which reduces the overall exploration speed.

6 THE RESTORE FRAMEWORK

In this section, we present a generalization of the *Restore* algorithm introduced by Wang et al. [2021]. Our formulation extends the method by allowing for state-dependent global dynamics and by permitting each local exploration to be driven by its own Markov process. In addition, we relax the theoretical assumptions required for correctness, enabling application to the light transport setting.

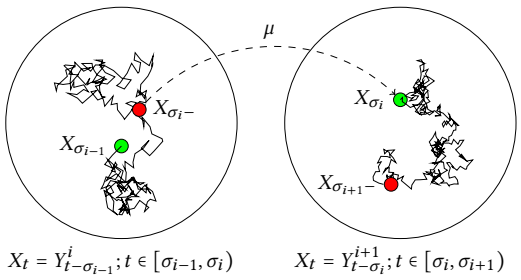

Fig. 2. Transfer between Y^i and Y^{i+1} using the transfer rule μ. The instance Y^i locally explores the space originating from $X_{\sigma_{i-1}}$. After termination, the algorithm jumps from the exit point X_{σ_i-} of the previous tour to the spawn location X_{σ_i} of the next tour Y^{i+1}.

Core idea. The core idea is to simulate a Markov process Y^1 - which does not need to be invariant with respect to the desired target distribution - for a finite time τ_1. The so-called *lifetime* of the simulation. A single simulation up to this lifetime is called a *tour* of the process. After the simulation has been terminated, the next tour of a (possibly, but not necessarily, different) Markov process Y^2 is spawned and simulation up to his lifetime τ_2. The spawn location of the next tour is drawn from a user-defined distribution $\mu_1\left(Y^1_{\tau_1-}, \cdot\right)$, where

$$Y^1_{\tau_1-} := \lim_{t \to \tau_1-} Y^1_t \tag{7}$$

denotes the *exit point* - that is, the last point being visited immediately before termination - of the previous tour. The transfer rule μ_1 can, but does not need to, depend on the exit point. By the transfer rule μ_1, the local exploration is transferred from one portion of the space to another. This is illustrated in Figure 2.

The Restore process. Continuing the idea above, the overall process is given by

$$X_t := Y^i_{t-\sigma_{i-1}} \quad \text{for } t \in [\sigma_{i-1}, \sigma_i), \tag{8}$$

where $\sigma_j := \sum_{i=1}^{n} \tau_i$ is the time elapsed after the jth instance has been executed. The spawn location of the ith tour is drawn from $\mu_{i-1}\left(Y^{i-1}_{\tau_{i-1}-}, \cdot\right)$.

6.1 MH as a special instance

MH can be viewed as a special case of the above construction. To this end, we revisit the interpretation outlined in section 1, according to which MH essentially simulates a (conceptually *local*) Markov chain Y, implicitly defined by the proposal rule.

Let σ_j denote the time of the jth rejection. Then, the MH chain naturally decomposes into tours as in (8), where the transfer rules μ_i are given by a Dirac kernel: upon rejection, MH restarts the Markov chain Y from its previous state.

Now consider specifically MH with a mixture proposal rule of the form (6). In this case, the interpretation of local and global dynamics can be embedded directly into our framework. Indeed: Let σ_j denote the time at which MH accepts the jth large step proposal. Then again, the MH chain decomposes into tours according to (8), but this time the transfer rules μ_i are all identical to the large step distribution μ. Consequently, the tours correspond to local explorations with finite lifetime, which are *killed* by the rejection step and globally *revived* at a new location according to μ.

6.2 Ensuring invariance

The concatenation of local processes Y^i after arbitrary lifetimes τ_i is, of course, not guaranteed to be invariant with respect to a desired target distribution π.

To obtain a parameterizable control mechanism over the lifetimes τ_i, we choose — following concepts from general theory as presented in Blumenthal and Getoor [1968] and Sharpe [1988] — to model the τ_i as clocks that decay with a time-dependent exponential rate

$$[0, \infty) \ni t \mapsto \gamma_i\left(Y^i_t\right). \tag{9}$$

Due to the interpretation of τ_i as lifetimes, the functions γ_i are referred to as *killing rates*.

As already observed in Wang et al. [2021], this assumption is sufficient to define a choice of γ_i that renders the concatenated process X invariant with respect to a desired target distribution π.

As in section 4, we assume that π admits a density p with respect to a reference measure λ. Under this assumption, the choice

$$\gamma_i := \frac{\left(L_i^* + c_i \mu_i^*\right)p}{p}, \tag{10}$$

where $c_i > 0$ must be chosen such that γ_i is well-defined ensures that the concatenated process X is invariant with respect to π.

The definition (10) requires further explanation. Here, L_i denotes the *generator* [Kallenberg 2021] of the Markov process Y^i. This is an operator that captures the infinitesimal behavior of the process and uniquely characterizes it. Formally, L_i acts on a function f via

$$(L_i f)(x) := \lim_{t \to 0+} \frac{\mathrm{E}\left[f\left(Y^i_t\right) \,\middle|\, Y^i_0 = x\right] - f(x)}{t}. \tag{11}$$

The transfer rules μ_i also act on functions f via

$$(\mu_i f)(x) := \int \mu(x, \mathrm{d}y) f(y). \tag{12}$$

We can thus formally view both L_i and μ_i as operators T on $L^2(\lambda)$ and consider their adjoints T^*, defined implicitly by the relation

$$\int gTf \,\mathrm{d}\lambda = \int fT^* g \,\mathrm{d}\lambda\,. \tag{13}$$

Definition 6.1. *Given the choice* (10) *of the killing rates γ_i, we refer to the concatenated process X as the* ***Restore process with local dynamics Y^i, global dynamics μ_i, and target distribution π.***

The choice (10) for the killing rates γ_i is intuitively plausible, as it is inversely proportional to the target density p. Accordingly, the killing rates γ_i are large in regions where the target density p is small and small where p is large.

The proof technique used in Wang et al. [2021] for the special case considered there required assumptions on the local process Y and the target density p that render the resulting procedure unsuitable for many applications — including light transport simulation. In the supplementary we provide a proof that applies in our, by far, more general setting and avoids such unnecessary restrictions.

Expected lifetime. For practical implementations — especially on GPUs — it is useful to observe that the only degree of freedom in the definition (10) of the killing rates γ_i is the constant c_i, which is precisely the expected lifetime τ_i of the ith tour.

Invariant local dynamics. If the local processes Y^i are already invariant with respect to the target distribution π, then

$$L_i^* p = 0; \tag{14}$$

see Kallenberg [2021]. In this case, the definition (10) of the killing rates γ_i simplifies considerably — especially since in practice, the operator L_i is often a computationally expensive integral operator.

State-independent global dynamics. If the transfer rules μ_i do not depend on the exit point of the previous tour — that is, if they are simple probability distributions — and if they admit a density u_i with respect to the same reference measure λ under which the target distribution π admits the density p, it is straightforward to verify

$$\mu_i^* g = u_i \int g \,\mathrm{d}\lambda\,. \tag{15}$$

If, additionally, (14) is satisfied, the killing rates simplify to

$$\gamma_i = c_i p_\lambda \frac{u_i}{p}. \tag{16}$$

User-defined local and global dynamics. To summarize: in the Restore framework, the user has three degrees of freedom:

(1) The local dynamics Y^i, which govern the local exploration within each tour.
(2) The global dynamics μ_i, which describe the transfer between successive local explorations (or the "rebirth locations" in the case of state-independent transfers).
(3) The expected lifetime c_i of the ith tour.

Practical implementation. The main difficulties in practical implementations lie in computing the term $L_i^* p$ in the definition (10) of the killing rate γ_i, and in simulating the lifetime τ_i itself. The former can be avoided by choosing local exploration processes Y^i that are already invariant with respect to the target distribution. The latter is more challenging. In the numerical study of this work, we avoid

this issue by focusing on a specific instance of the Restore process, which we describe in section 7, and for which a simple method for simulating τ_i exists.

Discussion. Compared to Wang et al. [2021], our formulation introduces several key generalizations. First, instead of using a single Markov process to govern the whole local dynamics, we allow each local exploration to follow its own Markov process. Second, we generalize the global dynamics: the starting point of the next local exploration can depend on the the exit point of the previous one. In contrast, Wang et al. [2021] used a fixed distribution for this step, which justified the interpretation of the algorithm as repeatedly *regenerating* a single Markov process. Finally, we relax the technical assumptions required for correctness, broadening the method's applicability — in particular, to domains such as light transport. We stress that correctness of our more general formulation can still be proven rigorously.

7 THE *JUMP* RESTORE ALGORITHM

To practically demonstrate the potential of the Restore framework presented in section 6, this work aims to show how any existing MCMC-based light transport algorithm can be made more efficient simply by integrating it into the Restore framework.

As described in section 5, existing approaches consist of local and global components of exploration. In this work, we empirically — and in the supplementary, theoretically — demonstrate that, given *any* MCMC-based light transport algorithm, using its local component as the local dynamics and its global component as the global dynamics of the Restore process, the resulting Restore process is more efficient than the baseline algorithm in its default form with both (local and global) components.

Since every existing MCMC-based light transport algorithm simulates a *discrete-time* Markov chain, while the Restore process — due to its use of time-dependent exponential killing rates — relies on a *continuous-time* formulation, we begin by describing how to embed a given discrete-time Markov chain into continuous time in such a way that its dynamics remain unaltered.

7.1 Embedding discrete-time Markov chains into continuous time

It is natural to embed a discrete-time chain into continuous time by *holding* the states for a (random) continuous duration. And indeed, as elementary results (as found in Kallenberg [2021]) show, using exponentially distributed holding times yields a time-homogeneous and Markovian process if the original chain was. If we choose the exponential distribution parameter to be 1, then the discrete-time Markov chain and the resulting continuous-time Markov process even share the same generator and thus follow the same dynamics.

Definition 7.1. *The process Y arising from the above embedding of a discrete-time chain M into continuous time is referred to as the* ***continuous-time embedding of M****.*

This yields the special case of the Restore framework introduced in section 6 that is central to our numerical study in section 9:

Definition 7.2. *Given a target distribution π and a transfer rule μ on the same space, the Restore process with local dynamics Y, global dynamics μ, and target distribution π is called the* ***jump Restore process with local dynamics M, global dynamics μ, and target distribution π****.*

7.2 Practical implementation

The key observation is that the jump Restore process — true to its name — is (like its local dynamics Y) a *pure-jump type* Markov process with transition rule

$$\big(1+\gamma(x)\big)^{-1}\zeta(x,\,\cdot\,)+\gamma(x)\big(1+\gamma(x)\big)^{-1}\mu \tag{17}$$

at current state x. Here, γ denotes *the* killing rate (10), which — due to the use of uniform local and global dynamics — does not depend on the tour index i, and ζ denotes the transition rule of the Markov chain M. The constant 1 in the numerators of (17) corresponds to our choice of exponential distribution parameter for the holding times.

Given this insight, the jump Restore process can be simulated just like any other pure-jump type Markov process:

Algorithm 7.1 Jump Restore algorithm
with local dynamics M, global dynamics μ and target distribution π

Input: Initial state x_0 and sample count $n \in \mathbb{N}$.
Output: Realization of the jump Restore process X

for $(i = 1;\,;\,{+}{+}i)$
  Sample t_1 from $\mathrm{Exp}(1)$; ← holding time of the current state x_{i-1}
  Sample t_2 from $\mathrm{Exp}\big(\gamma(x_{i-1})\big)$; ← time til next termination attempt
  if $(t_1 < t_2)$ ← next local state transition before termination attempt
    $\Delta t_i = t_1$;
    if $(i == n)$ **return** $\big((\Delta t_1, x_0), \ldots, (\Delta t_n, x_{n-1})\big)$;
    Sample x_i from $\zeta(x_{i-1},\,\cdot\,)$; ← local state transition
  else ← termination before next local state transition
    $\Delta t_i = t_2$;
    if $(i == n)$ **return** $\big((\Delta t_1, x_0), \ldots, (\Delta t_n, x_{n-1})\big)$;
    Sample x_i from $\mu(x_{i-1},\,\cdot\,)$; ← global state transition

The output of Algorithm 7.1 consists of the visited states paired with their respective holding times.

7.3 Estimation

Recall: To estimate the integral of a function f with respect to our target distribution π, we use (2) and compute

$$\frac{1}{t}\int_0^t f(X_s)\,\mathrm{d}s \approx \int f\,\mathrm{d}\pi \tag{18}$$

for sufficiently large $t > 0$. For simplicity, suppose that we simulate Algorithm 7.1 exactly until the completion of the jth tour — that is, $\sum_{i=1}^{j}\Delta t_i$ is a realization of σ_j. Then we obtain the practically usable approximation

$$\left(\sum_{i=1}^{n}\Delta t_i\right)^{-1}\sum_{i=1}^{n}\Delta t_i f(x_{i-1}) \approx \frac{1}{\sigma_j}\int_0^{\sigma_j} f(X_t)\,\mathrm{d}t \approx \int f\,\mathrm{d}\pi \tag{19}$$

using the left-hand rectangle rule for integral approximation.

8 PRACTICAL SETUP

As described in section 7, all existing MCMC-based light transport algorithms are variants of the MH algorithm, using a proposal kernel Q_ℓ of the mixture form (6), which combines local and global exploration components. We therefore assume that the reference algorithm against which we compare our method is given by an MH chain M^ℓ with proposal kernel Q_ℓ and target distribution π.

The key idea is to compare this reference algorithm with the jump Restore algorithm, which reuses the local component of the reference algorithm as the local dynamics and the global component as the global dynamics in the Restore framework. More precisely, we compare the reference algorithm M^ℓ with the jump Restore process X, which has local dynamics M^0, global dynamics μ, and target distribution π, as defined in Definition 7.2. Here, μ corresponds exactly to the large step distribution used in the definition of Q_ℓ.

Locally, the jump Restore process follows the same behavior as the local component of the reference algorithm, i.e., it evolves according to M^0. However, the length of each local exploration phase is controlled Restore's killing mechanism, which also ensures global discovery of the target distribution by triggering regeneration steps according to the global component μ of the reference algorithm.

9 NUMERICAL STUDY

In our evaluation, we compare three representative light transport algorithms: those proposed by Hachisuka et al. [2014], Luan et al. [2020a], and Li et al. [2015]. All three are ultimately grounded in classical MH algorithms, whose proposal kernels are derived from time-discretized diffusion processes — Brownian motion [Karatzas and Shreve 1998], Langevin dynamics, and Hamiltonian dynamics, respectively. While these light transport methods do not implement the corresponding statistical techniques in their original form — for example, all of them adopt the *multiplexing* technique introduced by Hachisuka et al. [2014] — they still follow the principles of their general-purpose analogues.

For clarity and consistency with the statistical foundations, we refer to these light transport algorithms as *Metropolis*, Metropolis-adjusted Langevin algorithm (MALA), and Hamiltonian Monte Carlo (H²MC) in this paper. Their jump Restore variants are denoted as *Metropolis Restore*, *MALA Restore*, and *H²MC Restore*, respectively.

Rendering environment. We implemented our method in the `pbrt` [Pharr et al. 2021] and `lmc` [Luan et al. 2020a] rendering system and applied the MCMC methods to both direct and indirect lighting.

Test scenes. We evaluated divserse scenes exhibiting different light transport characteristics. The scenes are from three sources: Contemporary Bathroom, Glass of Water, and Country Kitchen from Bitterli [2016]; Veach, Ajar and Torus from Luan et al. [2020b]; and Swimming Pool from Rioux-Lavoie et al. [2020].

Error metrics. We assessed several quantitative metrics: the L^1-error, L^2-error (i.e., mean squared error (MSE)), mean relative squared error (MRSE), relative mean squared error (RMSE), and mean absolute percentage error (MAPE). Reference images were generated using BDPT with 2^{20} samples per pixel (SPP). In addition, we computed the (empirical) variance of the resulting renderings.

Reference images were generated using BDPT with 2^{20} SPP. In addition, we computed the variance of the resulting renderings.

Evaluation. Qualitative comparisons are shown in figs. 1, 3, 5, 6 and 8, covering all test scenes introduced above - except the Country Kitchen scene, which can be found in the supplementary. They reveal that Restore variants consistently achieve better mode coverage and visual fidelity than their standard counterparts. Quantitative results in figs. 4 and 7 confirm this observation: both the L^2-error and empirical variance decrease more rapidly over time for the Restore variants. Detailed tables with absolute error values at equal time and equal SPP are provided in the supplementary.

9.1 Energy Redistribution Path Tracing (ERPT)

ERPT [Cline et al. 2005] shares conceptual similarities with the jump Restore algorithm proposed in this work. Both methods perform sampling through multiple, short chains that are initialized in different regions of the space and explore their local neighborhood.

More preciely, ERPT begins with a bootstrapping phase in which a specifiable number of random paths are generated and evaluated using traditional MC integration. Each path's contribution to the image — thought of as its "energy" in the ERPT context — is computed in this step. A subset of these paths is then selected as initial states for Markov chains, with selection biased toward higher-energy paths. These Markov chains then evolve by mutating the current path and "redistributing" its energy to the resulting path. The average number of started chains per pixel and a common fixed length for them are user-defined parameters.

Due to this conceptual similarity, we included ERPT in our numerical evaluation. From a theoretical perspective, however, ERPT exhibits some of the same challenges as traditional MH sampling: it may get trapped in local modes of the target distribution, and the choice of chain length and count is highly scene-dependent and difficult to tune universally. Additionally, since ERPT operates on a per-pixel basis, it is less closely related to our proposed algorithm than MH . We explicitly refer again to our discussion in subsection 6.1, which further clarifies how MH relates to our algorithm.

10 CONCLUSION

We introduced a generalized framework for MCMC sampling that overcomes key limitations of MH-based light transport algorithms. By decoupling local exploration and global discovery into separate mechanisms, our approach preserves desirable properties such as nonreversibility and rapid local exploration, while ensuring theoretical correctness and practical flexibility. This makes it possible to construct sampling schemes that explore efficiently, exploit local geometry or structure, and remain globally consistent — thereby bridging a longstanding gap between the needs of modern rendering applications and the limitations of classical MCMC methods. Beyond rendering, our framework holds strong potential for a broad spectrum of industrial applications, including generative AI, where the global discovery of multimodal distributions is also fundamental.

REFERENCES

Joris Bierkens. 2015. Non-reversible Metropolis-Hastings. *Statistics and Computing* 26, 6 (Aug. 2015), 1213–1228. https://doi.org/10.1007/s11222-015-9598-x

Benedikt Bitterli. 2016. Rendering resources. https://benedikt-bitterli.me/resources/.

Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2017. Reversible jump Metropolis light transport using inverse mappings. *ACM Transactions on Graphics* 37, 1 (oct 2017).

Robert McCallum Blumenthal and Ronald Kay Getoor. 1968. *Markov Processes and Potential Theory*. Vol. 29. Academic Press.

David Cline, Justin Talbot, and Parris Egbert. 2005. Energy redistribution path tracing. *ACM Trans. Graph.* 24, 3 (July 2005), 1186–1195. https://doi.org/10.1145/1073204.1073330

Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth. 1987. Hybrid Monte Carlo. *Physics Letters B* 195, 2 (1987).

Stewart N. Ethier and Thomas G. Kurtz. 2009. *Markov Processes: Characterization and Convergence*. John Wiley & Sons.

P.J. Green. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82, 4 (1995).

Toshiya Hachisuka, Anton S. Kaplanyan, and Carsten Dachsbacher. 2014. Multiplexed Metropolis light transport. *ACM Transactions on Graphics* 33, 4 (2014).

W. K. Hastings. 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57, 1 (4 1970).

Olav Kallenberg. 2021. *Foundations of Modern Probability* (3 ed.). Springer Nature Switzerland AG 2021.

Ioannis Karatzas and Steven E. Shreve. 1998. *Brownian Motion and Stochastic Calculus* (2 ed.). Springer New York, NY.

Csaba Kelemen, László Szirmay-Kalos, György Antal, and Ferenc Csonka. 2002. A simple and robust mutation strategy for the Metropolis light transport algorithm. *Computer Graphics Forum* 21, 3 (2002).

Tzu-Mao Li, Jaakko Lehtinen, Ravi Ramamoorthi, Wenzel Jakob, and Frédo Durand. 2015. Anisotropic gaussian mutations for Metropolis light transport through Hessian-Hamiltonian dynamics. *ACM Transactions on Graphics* 34, 6 (2015).

J. S. Liu, F. Liang, and W. H. Wong. 2000. The multiple-try method and local optimization in Metropolis sampling. *J. Amer. Statist. Assoc.* 95, 449 (2000).

Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020a. Langevin Monte Carlo rendering with gradient-based adaptation. *ACM Transactions on Graphics* 39, 4 (2020).

Fujun Luan, Shuang Zhao, Kavita Bala, and Ioannis Gkioulekas. 2020b. *lmc*. https://github.com/luanfujun/Langevin-MCMC

E Marinari and G Parisi. 1992. Simulated Tempering: A New Monte Carlo Scheme. *Europhysics Letters (EPL)* 19, 6 (July 1992).

Hector McKimm, Andi Q. Wang, Murray Pollock, Christian P. Robert, and Gareth O. Roberts. 2024. Sampling using Adaptive Regenerative Processes. arXiv:2210.09901 https://arxiv.org/abs/2210.09901

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21, 6 (6 1953).

S.P. Meyn and R.L. Tweedie. 1993. *Markov Chains and Stochastic Stability*. Springer-Verlag, London. /brokenurl#probability.ca/MT

Antonietta Mira. 2011. On Metropolis-Hastings algorithms with delayed rejection. *Journal of Statistics* (2011).

Jacopo Pantaleoni. 2017. Charted Metropolis light transport. *ACM Transactions on Graphics* 36, 4 (jul 2017).

Matt Pharr, Wenzel Jakob, and Greg Humphreys. 2021. *Physically Based Rendering, fourth edition*. The MIT Press. https://pbr-book.org/

Damien Rioux-Lavoie, Joey Litalien, Adrien Gruson, Toshiya Hachisuka, and Derek Nowrouzezahrai. 2020. Delayed rejection Metropolis light transport. *ACM Transactions on Graphics* 39, 3 (may 2020). https://doi.org/10.1145/3388538

Gareth O. Roberts and Richard L. Tweedie. 1996. Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli* 2, 4 (1996).

Benjamin Segovia, Jean Claude Iehl, and Bernard Péroche. 2007. Coherent Metropolis light transport with multiple-try mutations. https://api.semanticscholar.org/CorpusID:18709785

Michael Sharpe. 1988. *General Theory of Markov Processes*. Vol. 133. Academic Press.

Eric Veach. 1997. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph. D. Dissertation. Stanford University.

Andi Q. Wang. 2020. *Theory of killing and regeneration in continuous-time Monte Carlo sampling*. Ph. D. Dissertation. University of Oxford.

Andi Q. Wang, Murray Pollock, Gareth O. Roberts, and David Steinsaltz. 2021. Regeneration-enriched Markov processes with application to Monte Carlo. *The Annals of Applied Probability* 31, 2 (2021).

Erhan Çinlar. 2011. *Probability and Stochastics*. Springer Science+Business Media.

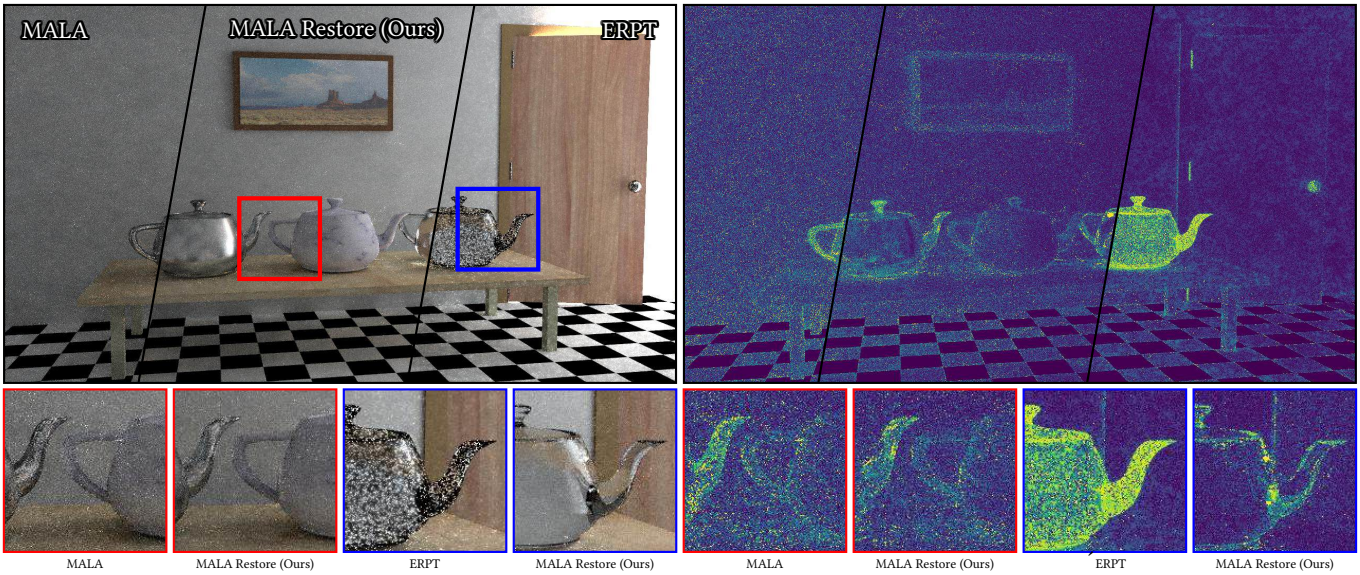

Fig. 3. Equal rendering time comparison (20 s) of MALA (left), MALA Restore (middle), and ERPT (right) for the Veach, ajar scene provided by [Luan et al. 2020b].

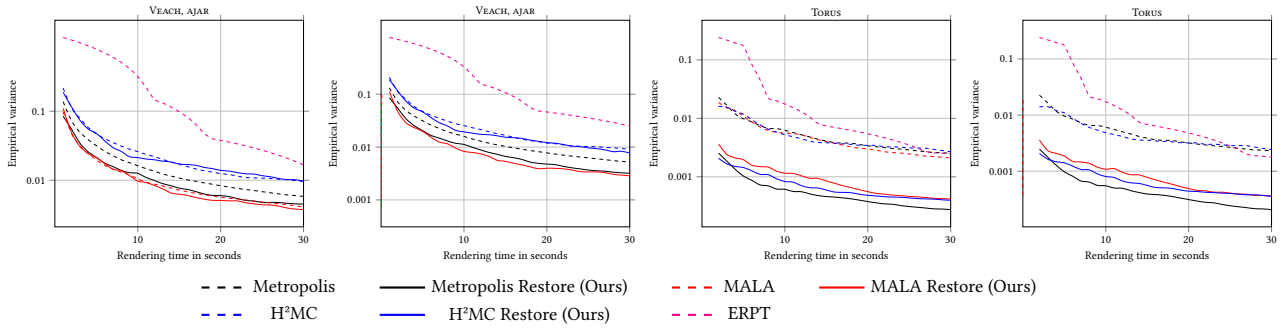

Fig. 4. L^2-error and empirical variance over rendering time in seconds for the Veach, ajar and Torus scene depicted in Figure 3 and Figure 5, respectively.

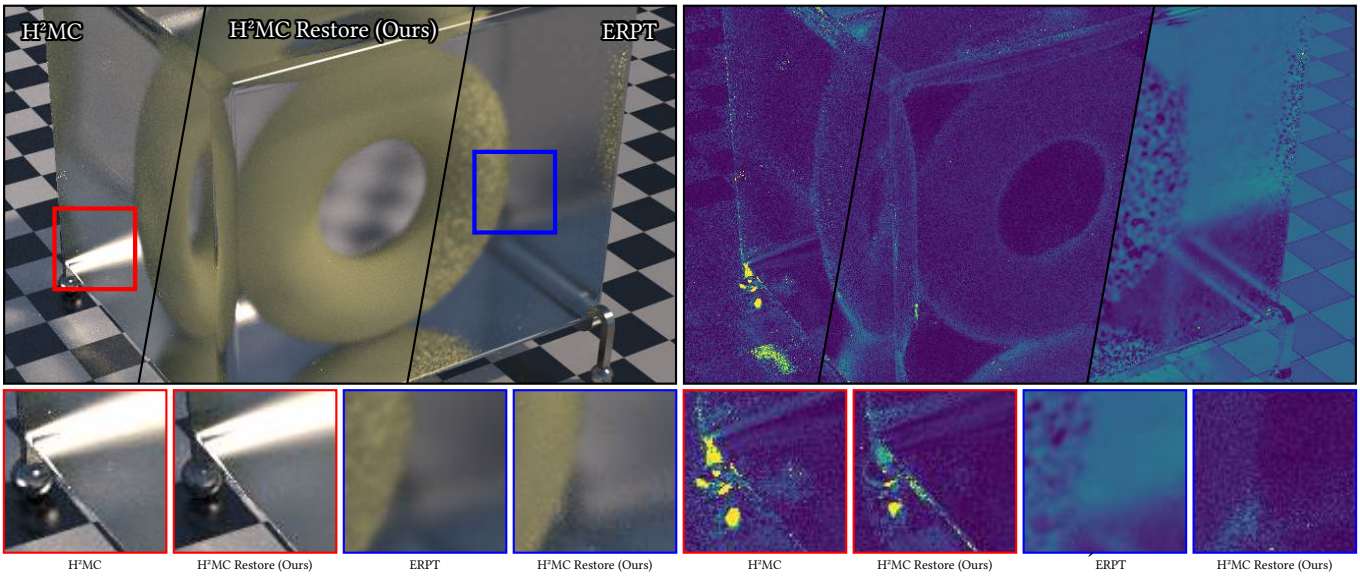

Fig. 5. Equal rendering time comparison (120 s) of H²MC (left), H²MC Restore (middle), and ERPT (right) for the Torus scene provided by [Luan et al. 2020b].

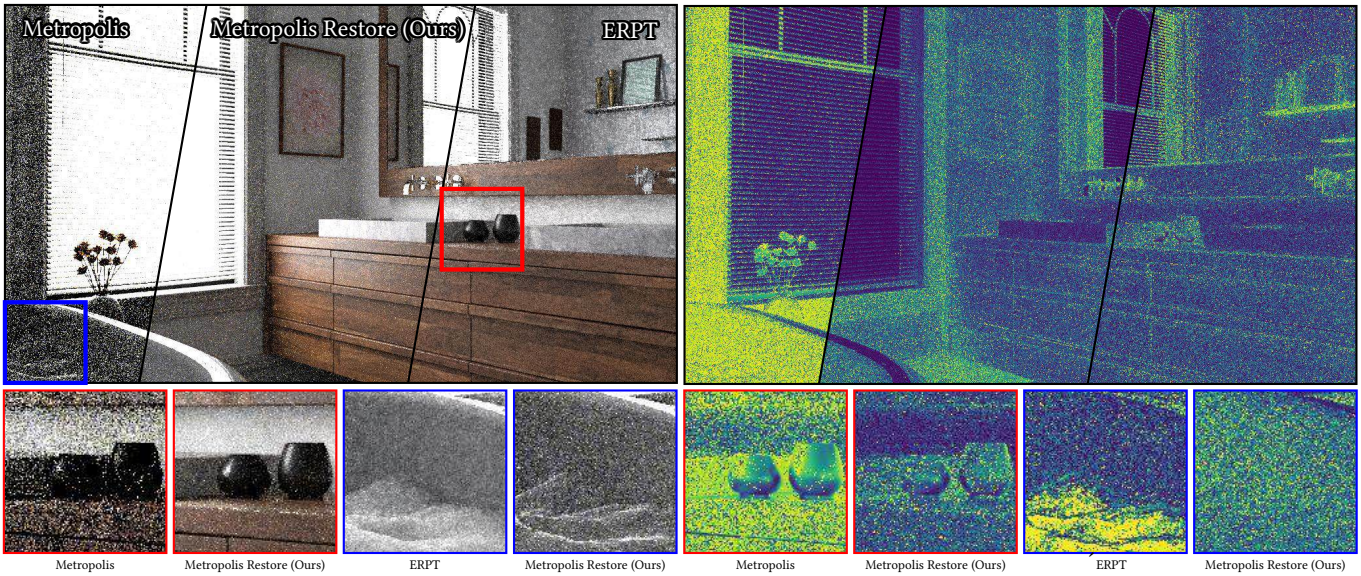

Fig. 6. Equal rendering time comparison (120 s) of Metropolis (left), Metropolis Restore (middle), and ERPT (right) for the Contemporary Bathroom scene provided by [Bitterli 2016].

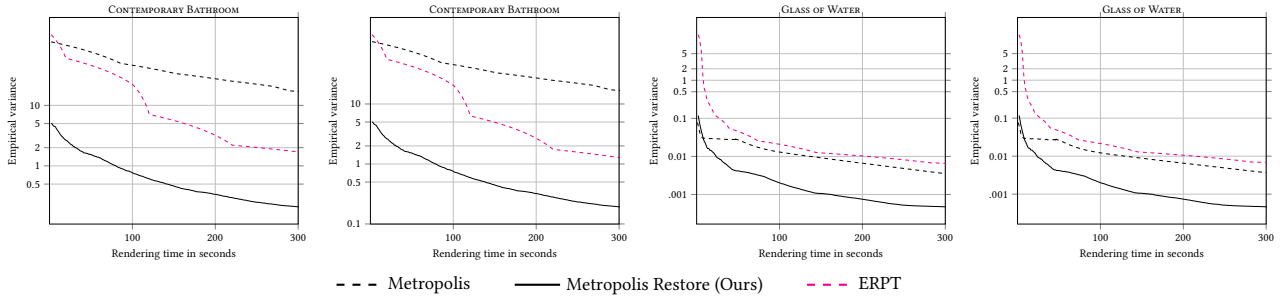

Fig. 7. L^2-error and empirical variance over rendering time in seconds for the Veach, ajar and Torus scene depicted in Figure 3 and Figure 5, respectively.

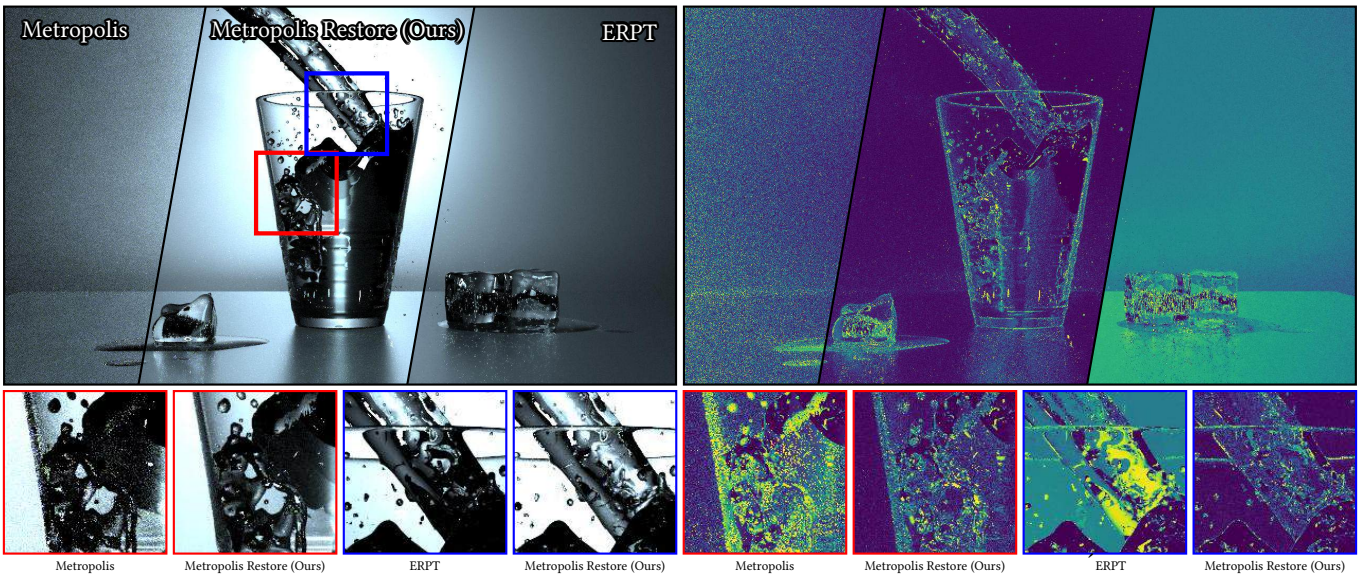

Fig. 8. Equal rendering time comparison (60 s) of Metropolis (left), Metropolis Restore (middle), and ERPT (right) for the Glass of Water scene provided by [Bitterli 2016].