Bayesian delensing of CMB temperature and polarization

Marius Millea, Ethan Anderes, Benjamin D. Wandelt

Bayesian delensing of CMB temperature and polarization

Marius Millea,[1,2,*] Ethan Anderes,[3] and Benjamin D. Wandelt[1,2,4,5]

[1]*Institut d'Astrophysique de Paris (IAP), UMR 7095, CNRS UPMC Universit Paris 6, Sorbonne Universités, 98bis boulevard Arago, F-75014 Paris, France*

[2]*Institut Lagrange de Paris (ILP), Sorbonne Universités, 98bis boulevard Arago, F-75014 Paris, France*

[3]*Department of Statistics, University of California, Davis, California 95616, USA*

[4]*Department of Physics and Astronomy, University of Illinois at Urbana-Champaign, 1002 W Green St, Urbana, Illinois 61801, USA*

[5]*Center for Computational Astrophysics, Flatiron Institute, 162 5th Avenue, 10010, New York, New York, USA*

(Received 23 August 2017; revised manuscript received 26 March 2019; published 10 July 2019)

We develop the first algorithm able to jointly compute the maximum *a posteriori* estimate of the Cosmic Microwave Background (CMB) temperature and polarization fields, the gravitational potential by which they are lensed, and the tensor-to-scalar ratio, r. This is an important step towards sampling from the joint posterior probability function of these quantities, which, assuming Gaussianity of the CMB fields and lensing potential, contains all available cosmological information and would yield theoretically optimal constraints. Attaining such optimal constraints will be crucial for next-generation CMB surveys like CMB-S4, where limits on r could be improved by factors of a few over currently used suboptimal quadratic estimators. The maximization procedure described here depends on a newly developed lensing algorithm, which we term LenseFlow, and which lenses a map by solving a system of ordinary differential equations. This description has conceptual advantages, such as allowing us to give a simple nonperturbative proof that the determinant of LenseFlow on pixelized maps is equal to unity, which is crucial for our purposes and unique to LenseFlow as compared to other lensing algorithms we have tested. It also has other useful properties such as that it can be trivially inverted (i.e., delensing) for the same computational cost as the forward operation, and can be used to compute lensing adjoint, Jacobian, and Hessian operators. We test and validate the maximization procedure on flat-sky simulations covering up to 600 deg^2 with nonwhite noise and masking.

DOI: 10.1103/PhysRevD.100.023509

I. INTRODUCTION

Weak gravitational lensing of the Cosmic Microwave Background (CMB) by an intervening large scale structure plays and will continue to play a crucial role in the ability of cosmological observations to constrain fundamental physics. For example, the gravitational lensing effect already allows a completely independent confirmation of the existence of dark energy from the CMB alone [1], and future experiments such as CMB-S4 are predicted to map out the gravitational lensing potential field, ϕ, precisely enough to measure for the first time the absolute neutrino mass scale and potentially differentiate the two possible mass hierarchies [2]. A wealth of cosmological and astrophysical information can also be extracted from these lensing potential maps in cross correlation with other datasets (see, e.g., [3]).

The most profound impact from CMB lensing on our understanding of the Universe, however, may come not from measuring the effect, *per se*, but rather from our ability to remove it. Lensing aliases E-mode polarization into B-modes, which can obscure the primordial B signal expected to come from gravitational waves produced during inflation. Due to its unique signature, it is possible to undo the lensing effect, a process usually called "delensing". This will be crucial to placing the tightest possible constraints on the amplitude, r, of the gravitational wave B-modes. If detected, the primordial signal would offer an unprecedented window into the extremely early Universe and to energy scales impossible to probe with terrestrial particle accelerators.

Delensing of both T and E can also be useful as it leads to a sharpening of the acoustic peaks. This in turn makes it easier to measure their phase and could lead to detecting or ruling out the presence of extra species of relativistic particles in the Universe [4].

Despite the important role delensing is expected to play in future CMB constraints, currently no workable fully optimal delensing algorithm exists. To date, all delensing analyses on real data have been based on a quadratic

*Corresponding author.
mariusmillea@gmail.com

estimate of the lensing potential [5,6]. While the quadratic estimator is nearly optimal at current noise levels, it will become significantly suboptimal once noise levels cross below the ~ 5 μK-arcmin effective noise level of the lensing contribution (exactly when delensing becomes most important). The suboptimality of the quadratic estimate stems from the fact that the total B-mode power is a source of noise for the estimator, meaning the results can be improved by repeatedly using the lensing potential estimate to delense the data and then reestimating the lensing potential. Such iterative delensing algorithms have been discussed in some form in, e.g., [7–10].

Two concrete iterative delensing examples which can be considered precursors to our work have been given by [11,12]. In a similar manner to iterating a quadratic estimate, both of these algorithms iteratively maximize the Bayesian posterior probability $\mathcal{P}(\phi|d,r)$, where ϕ is the lensing potential and d is the CMB temperature and polarization data.[1] In terms of the end product, the two differ largely in that the latter algorithm computes the exact maximum and was demonstrated to be robust even in the presence of masking.

These works greatly improve the optimality of the lensing reconstruction and represent key advances in CMB lensing analysis. However, neither estimate is exactly optimal (the posterior mean of ϕ being the optimal estimate with respect to the mean squared error), and neither readily produces an estimate of an unlensed map nor of r. Indeed, since the temperature and polarization fields themselves are implicitly marginalized over in $\mathcal{P}(\phi|d,r)$, unlensed fields are not estimated at all by these procedures. The resulting best-fit ϕ could be used to delense the data, but as we will discuss, this resulting delensed data does not have any Bayesian interpretation. The delensed map could be taken as an estimator, but would still require simulations to debias and quantify uncertainty, similarly as for the quadratic estimate but with a more costly procedure to simulate. More importantly, it is not entirely clear how to do this at all because these simulations would depend on r, the quantity we are trying to estimate in the first place. Indeed, in their stated form both algorithms take r as given, rather than jointly estimating it or marginalizing over it.

A conceptually straightforward solution to these issues which would yield optimal constraints on all of these quantities is to obtain samples from the *joint* Bayesian posterior probability function, $\mathcal{P}(f,\phi,r|d)$, including both the unlensed fields, $f \equiv (T,Q,U)$, and the tensor-to-scalar ratio, r. Here, we present the first algorithm which is able to efficiently *maximize* this probability distribution, an important advancement towards the ultimate goal of obtaining samples. Additionally, the best fit computed here can be used as an initialization for a sampler, and we expect that a good starting point will be important due to the high dimensionality of the problem (the number of dimensions here being the number of map pixels, which can be in the millions). Although we do not expect joint sampling to be without challenges, it has already been demonstrated on temperature-only data by [13], and we view the techniques developed here as having solved the more difficult aspects of the problem of extending to polarization. We leave full discussion of sampling with temperature and polarization for a follow-up work, here discussing mainly maximization.

The results here also differ from [13] by exploring r as a free parameter. In some sense it is quite easy to maximize over r, since we can trivially parallelize the maximization over f and ϕ across a grid of r values. Doing so, we will show that the maximum *a posteriori* (MAP) estimate of r in this joint case is always zero. Thus, while we demonstrate we are able to maximize the posterior over r, the resulting best fit is not useful for cosmological parameter inference. We will thus focus most of our discussion on $\mathcal{P}(f,\phi|d,r)$.

As opposed to exact maximization of $\mathcal{P}(\phi|d,r)$ which was solved by [12], maximization of $\mathcal{P}(f,\phi|d,r)$ is more difficult not just because of the increased dimensionality of the problem, but because f is highly correlated with ϕ. Intuitively, this is simply because an observed hot spot at some position could be a true hot spot there with no lensing, or a nearby hot spot deflected to that position by lensing. This degeneracy leads to extremely slow convergence unless the correlations are carefully taken into account. We find an advantageous way to do so is to reparametrize the posterior probability function in terms of the lensed fields (denoted by $\tilde{f}$) instead of the of the unlensed ones, similarly as in [13]. This greatly reduces the correlations, but the change of variables introduces a complicated determinant term in the posterior probability which depends on ϕ. Having to calculate this quantity might render the reparametrization ultimately useless in practice. However, we are able to propose and validate an approximation to this reparametrized posterior which does not contain such a complicated determinant term. The proof relies on a new and accurate pixelized lensing approximation which we have developed called LenseFlow, which has the crucial property of having the unit determinant.

We use this in a maximization algorithm that can be regarded as an approximate coordinate descent, meaning we alternate updating $\tilde{f}$ with ϕ held constant then updating ϕ with the $\tilde{f}$ held constant. The former step amounts to a straightforward Wiener filter, and the latter step can be approximated with a quasi Newton-Raphson step. As we will show, a fundamental advantage of the lensed parametrization (in addition to reducing correlations), is that it

[1]These algorithms actually produce estimates of the full lensing displacement vector field, not just of ϕ which gives only the curl-free part in the Helmholtz decomposition of the displacement. For simplicity, we will ignore the divergence-free component throughout this work as it is expected to be too small to significantly impact the ϕ reconstruction at CMB-S4 noise levels [11], but it is straightforward to introduce it in our equations alongside ϕ.

removes all explicit dependence on data or instrument from this latter step. These two steps are repeated until convergence to the exact joint posterior maximum, which, depending on the exact data configuration and complexity of masking, we can achieve in 30 minutes to tens of hours on a single multicore CPU for maps as large as ∼600 deg^2 (with 3 arcmin pixels).

By contrast, the maximization procedure described in [12] is computationally much more costly due to the calculation of a determinant gradient term. We will discuss why our seemingly complicating addition of jointly estimating f actually makes the problem computationally easier, and what the trade-off has been in not computing this determinant. Furthermore, we will argue that even if one was only interested in posterior samples of r, it will still be computationally simpler to obtain them by sampling the joint posterior rather than the one marginalized over f.

The maximization makes use of exact posterior gradients, which are computable with LenseFlow. We show that even though Hessians of the posterior can not be stored in practice, their action on vectors can be efficiently calculated, a fact which is perhaps not widely appreciated. Although we do not use them here, Hessians could be quite beneficial to sampling algorithms.

Our code is available publicly.[2] It is written in the Julia programming language [14], making it fast while maintaining flexibility and readability. The link also contains a Jupyter notebook with a 128×128 pixel maximization example which completes in around two minutes on a modern laptop.

We begin the paper by deriving the joint Bayesian posterior in Sec. II and discussing how it is related to the marginalized posterior in Sec. II A. We then derive the coordinate descent equations for the joint posterior maximization in Sec. III. We develop LenseFlow, its gradients, as well as the proof that its determinant is unity in Sec. IV. We show results on simulated data in Sec. V. The results are broken into several parts for clarity of presentation, first with only Fourier-space masking in Sec. V A, next with map-level masking as well in Sec. V B, and then with r included as a free parameter in Sec. V C. Finally, we revisit and validate a posterior approximation (used in earlier sections) in Sec. VI and numerically verify the determinants of some lensing algorithms in Sec. VII.

II. THE JOINT POSTERIOR PROBABILITY

To start, we derive the target probability function that we seek to maximize in this work, mainly the joint posterior probability of the unlensed CMB, the CMB gravitational lensing potential, and the cosmological parameters.

Briefly summarizing our notation, we use ϕ for the gravitational lensing potential and f to describe a CMB field such as the temperature, T, or a tuple including polarization Stokes parameters, such as (Q, U) or (T, Q, U). Lensed fields are denoted with a tilde, $\tilde{f}$. Quantities like $\tilde{f}$, f, or ϕ should be thought of as abstract vectors, meaning they can be added and scaled without need to reference the basis in which they are represented. Indeed, most of our equations are written without reference to basis; at the few points where it *is* necessary to do so, we use $f(x)$ or $f(l)$ to refer to the real-space or Fourier basis. We use the notation $f^\dagger g$ to denote the inner product between fields f and g, which is defined to be a sum over products of corresponding temperature and polarization pixels in f and g. Linear operators on this resulting Hilbert space will be "blackboard" characters, e.g., $\mathbb{L}$, and adjoint operators, $\mathbb{L}^\dagger$, are defined as usual by the property that $f^\dagger(\mathbb{L}g) = (\mathbb{L}^\dagger f)^\dagger g$ for all f and g. We often use $\mathbb{L}^{-\dagger}$ as shorthand for the inverse then adjoint of the operator.

We model the data as,

$$d = \mathbb{P}\mathbb{L}(\phi) f + n. \tag{1}$$

Here, $\mathbb{L}(\phi)$ is the lensing operation, $\mathbb{P}$ is a pixelization operator which takes the infinite resolution lensed field, $\tilde{f} \equiv \mathbb{L}(\phi) f$, and pixelizes it down to the data resolution, and n is the noise contribution. For now, we neglect explicitly writing the beam, instrumental transfer functions, and masking, although they can straightforwardly be included by considering them as part of the $\mathbb{P}$ operator. We include these effects in our simulations and corresponding methodology later in Sec. V.

Assuming the noise is a Gaussian random field with covariance $\mathbb{C}_n$, the likelihood of the data is, up to an irrelevant normalization constant,

$$\begin{aligned} &-2\log\mathcal{P}(d|f,\phi) \\ &\quad = [d - \mathbb{P}\mathbb{L}(\phi) f]^\dagger \mathbb{C}_n^{-1} [d - \mathbb{P}\mathbb{L}(\phi) f]. \end{aligned} \tag{2}$$

By Bayes theorem, the posterior probability of f, ϕ, and of any cosmological parameters, θ, is proportional to this likelihood times a prior $\mathcal{P}(f, \phi, \theta)$,

$$\begin{aligned} &-2\log\mathcal{P}(f,\phi,\theta|d) = -2\log\mathcal{P}(d|f,\phi) - 2\log\mathcal{P}(f,\phi,\theta) \\ &= [d - \mathbb{P}\mathbb{L}(\phi) f]^\dagger \mathbb{C}_n^{-1} [d - \mathbb{P}\mathbb{L}(\phi) f] \\ &\quad + f^\dagger \mathbb{C}_f(\theta)^{-1} f + \log\det\mathbb{C}_f(\theta) \\ &\quad + \phi^\dagger \mathbb{C}_\phi(\theta)^{-1} \phi + \log\det\mathbb{C}_\phi(\theta). \end{aligned} \tag{3}$$

One is entirely free to chose the prior function to be as informative or uninformative as desired, although *something* about f must be specified for a posterior constraint of ϕ to be produced. Here we adopt the prior that both f and ϕ are independent Gaussian random fields with covariance given by $\mathbb{C}_f$ and $\mathbb{C}_\phi$, respectively, each of which may depend on some set of cosmological parameters, θ. We ignore any

[2] https://www.github.com/marius311/CMBLensing.jl.

prior correlation between f and ϕ, the most dominant expected contribution being at large scales in temperature due to the late-time integrated Sachs-Wolfe effect. It is straightforward to include this in (3), but we have not done so for simplicity and since it is likely too small to matter at the scales probed by the patches of sky considered here. Additionally, as mentioned in [12], using a Gaussian prior on ϕ (and in our case, f) does not outright erase from the reconstruction any non-Gaussianities that may be present in f and/or ϕ from various higher order effects. However, it does mean the posterior itself is formally incorrect if non-Gaussianities exist, since it incorporates a prior that assumes otherwise; the correct way to include them would be to forward model them in some form as part of the prior. At noise levels achievable in the near future, these non-Gaussianities must be accounted for [15,16], although how exactly to construct the necessary prior term is an open question and beyond the scope of this work.

Equation (3) is the posterior probability in terms of the infinite resolution unlensed field, f. What we actually work with in this paper differs in two ways. First, there is much less posterior correlation between $\tilde{f}$ and ϕ than there is between f and ϕ, and it is in fact a necessity to work with the former parametrization; otherwise, sampling and maximizing becomes prohibitively difficult. Second, we are, of course, estimating the lensed field on some pixels, a quantity which we will denote by $\tilde{f}_p \equiv \mathbb{P}\mathbb{L}(\phi)f$. Noting that the prior distribution of $\tilde{f}_p$ at fixed ϕ remains Gaussian with covariance $\Sigma_{\tilde{f}_p} \equiv \mathbb{P}\mathbb{L}(\phi)\mathbb{C}_f(\theta)\mathbb{L}(\phi)^\dagger\mathbb{P}^\dagger$, its posterior distribution is similarly straightforward to write from Bayes theorem,

$$\begin{aligned}
&-2\log\mathcal{P}(\tilde{f}_p,\phi,\theta|d)\\
&=-2\log\mathcal{P}(d|\tilde{f}_p,\phi)-2\log\mathcal{P}(\tilde{f}_p,\phi,\theta)\\
&=(d-\tilde{f}_p)^\dagger\mathbb{C}_n^{-1}(d-\tilde{f}_p)\\
&\quad+\tilde{f}_p^\dagger\Sigma_{\tilde{f}_p}^{-1}(\theta,\phi)\tilde{f}_p+\log\det\Sigma_{\tilde{f}_p}(\theta,\phi)\\
&\quad+\phi^\dagger\mathbb{C}_\phi(\theta)^{-1}\phi+\log\det\mathbb{C}_\phi(\theta).
\end{aligned}\tag{4}$$

At first glance, this seems like a difficult posterior to work with, because $\Sigma_{\tilde{f}_p}$ depends on ϕ and is not diagonal in any simple basis but must both be inverted and have its determinant calculated. We solve this problem by proposing the following approximation:

$$\begin{aligned}
&-2\log\mathcal{P}(\tilde{f}_p,\phi,\theta|d)\\
&\approx(d-\tilde{f}_p)^\dagger\mathbb{C}_n^{-1}(d-\tilde{f}_p)\\
&\quad+\tilde{f}_p^\dagger\mathbb{L}_p^{-\dagger}(\phi)\mathbb{C}_{f_p}(\theta)\mathbb{L}_p^{-1}(\phi)\tilde{f}_p+\log\det\mathbb{C}_{f_p}(\theta)\\
&\quad+\phi^\dagger\mathbb{C}_\phi(\theta)^{-1}\phi+\log\det\mathbb{C}_\phi(\theta),
\end{aligned}\tag{5}$$

where $\mathbb{L}_p$ is some pixelized lensing approximation (note the difference between $\mathbb{L}_p$ and $\mathbb{L}$, the latter which refers to the true lensing operation on infinite resolution). In Sec. VII, we present a method for proving that this approximation holds for some given experimental configuration. This proof rests on developing a new pixelized lensing algorithm, LenseFlow (Sec. IV), which has the property that $\log\det\mathbb{L}_p(\phi)=0$ for any finite pixel resolution. Using this method, we show that the approximation holds for CMB-S4-type configurations if we use *LenseFlow* itself as $\mathbb{L}_p$ in (5). Although we have not checked, it is possible this method can be used to show other lensing approximations would work as well.

We will thus work with (5) in the rest of this work, and for clarity drop the subscript p.

A. Relation to marginalized posteriors

In studies where the parameter of interest is ϕ, one may integrate out the unknown f to obtain the marginal posterior given by

$$\mathcal{P}(\phi|d)=\int \mathrm{d}f\,\mathcal{P}(f,\phi|d)\tag{6}$$

(we will drop explicitly labeling θ in this section).

This integral can be done analytically, and it is this probability distribution which is maximized by the algorithms given in [11,12]. In this section we compare the differences between this marginal estimate and the one developed here which maximizes the joint $\mathcal{P}(f,\phi|d)$.

The analytic marginalization over f can be regarded as an application of the Laplacian approximation method, which is exact in this case due to the Gaussianity of $\mathcal{P}(f|\phi,d)$, and which we give here since it also helps clarify the differences between the two estimates and the algorithms for computing them. To derive the Laplace approximation, first notice that for any fixed ϕ the function $f\mapsto\log\mathcal{P}(f,\phi|d)$ is quadratic in f. This implies there exists a normalization, $Z(\phi)$, which makes $f\mapsto\mathcal{P}(f,\phi|d)/Z(\phi)$ a Gaussian probability measure. In particular there exists $\hat{f}(\phi)$ and $\Sigma(\phi)$ such that, up to a constant,

$$\begin{aligned}
&-2\log[\mathcal{P}(f,\phi|d)/Z(\phi)]\\
&=[f-\hat{f}(\phi)]^\dagger\Sigma(\phi)^{-1}[f-\hat{f}(\phi)]+\log\det\Sigma(\phi),
\end{aligned}\tag{7}$$

where $\hat{f}(\phi)=\mathrm{argmax}_f\log\mathcal{P}(f,\phi|d)$ and $\Sigma(\phi)$ is the negative inverse Hessian of $f\mapsto\log\mathcal{P}(f,\phi|d)$. One can explicitly compute $\Sigma(\phi)$, $\hat{f}(\phi)$ and $Z(\phi)$ as follows:

$$\Sigma(\phi)=[\mathbb{L}(\phi)^\dagger\mathbb{C}_n^{-1}\mathbb{L}(\phi)+\mathbb{C}_f^{-1}]^{-1}\tag{8}$$

$$\hat{f}(\phi)=\Sigma(\phi)\mathbb{L}(\phi)^\dagger\mathbb{C}_n^{-1}d\tag{9}$$

$$Z(\phi)=\det\Sigma(\phi)^{\frac{1}{2}}\mathcal{P}(\hat{f}(\phi),\phi|d).\tag{10}$$

By multiplying and dividing $Z(\phi)$ in (6), while using the fact that $\mathcal{P}(f,\phi|d)/Z(\phi)$ integrates to 1 over f, the marginal posterior over ϕ is then given by

$$\mathcal{P}(\phi|d) = \det\Sigma(\phi)^{\frac{1}{2}}\mathcal{P}(\hat{f}(\phi),\phi|d) \propto \frac{\mathcal{P}(\hat{f}(\phi),\phi|d)}{\det[\mathbb{L}(\phi)\mathbb{C}_f\mathbb{L}(\phi)^\dagger + \mathbb{C}_n]^{\frac{1}{2}}}. \quad (11)$$

Equation (11) thus shows the marginal posterior on ϕ in the form of the Laplace approximation.

Now, to distinguish marginal versus joint MAP estimates we set the following notation:

$$\hat{\phi}_M \equiv \operatorname*{argmax}_\phi \mathcal{P}(\phi|d) \quad (12)$$

$$\hat{\phi}_J \equiv \operatorname*{argmax}_\phi \mathcal{P}(\hat{f}(\phi),\phi|d) \quad (13)$$

$$\hat{f}_M \equiv \hat{f}(\hat{\phi}_M) \quad \text{and} \quad \hat{f}_J \equiv \hat{f}(\hat{\phi}_J), \quad (14)$$

where $\hat{\phi}_M$ corresponds to the marginal estimate of ϕ and

$$(\hat{\phi}_J,\hat{f}_J) = \operatorname*{argmax}_{\phi,f} \mathcal{P}(f,\phi|d)$$

corresponds to the joint MAP estimate of both ϕ and f.

First notice that $\hat{\phi}_M$ and $\hat{\phi}_J$ are maximizing nontrivially different objectives, (12) versus (13), so clearly $\hat{\phi}_J \neq \hat{\phi}_M$ and hence $\hat{f}_J \neq \hat{f}_M$ as well. The fact that these estimates are different is an explicit manifestation of the non-Gaussianity of the posterior $\mathcal{P}(f,\phi|d)$, for otherwise marginal and joint MAP estimates would agree. More importantly, however, $\hat{f}_M$ can not be interpreted as a MAP estimate of the CMB, but rather as an intermediate variable used for the Laplace approximation technique of marginalization. This is not to say that $\hat{f}_M$ could not have reasonable sampling properties as a statistical estimator, but rather that $\hat{f}_M$ does not have an interpretation in the Bayesian framework.

In Sec. III we present an iterative algorithm for computing $(\hat{\phi}_J,\hat{f}_J)$ which shares some similarities to the one given in [12] for computing $\hat{\phi}_M$. However, the similarities are largely superficial. While both algorithms do generate a sequence of iterations …, (f_i,ϕ_i), … where f_i is defined recursively by a generalized Wiener filter of the unlensed CMB given the previous ϕ_{i-1}, i.e., $f_i = \hat{f}(\phi_{i-1})$, important differences arise in how ϕ_i is computed. In [12] the update ϕ_i is computed as the solution to a stationary equation characterizing the maximum of (11) with f_i in place of $\hat{f}(\phi)$. In contrast, the algorithm given in Sec. III updates ϕ_i using the lensed CMB parametrization $(\phi,\tilde{f})$ and, as such, is computed as an approximate maximizer of the *lensed* posterior given $\tilde{f}_i = \mathbb{L}(\phi_{i-1})f_i = \mathbb{L}(\phi_{i-1})\hat{f}(\phi_{i-1})$. In particular,

$$\phi_i \approx \operatorname*{argmax}_\phi \mathcal{P}(\mathbb{L}(\phi)^{-1}\tilde{f}_i,\phi|d). \quad (15)$$

One way to see the impact of this difference is through the data term $-\frac{1}{2}(d-\tilde{f}_i)^\dagger\mathbb{C}_n^{-1}(d-\tilde{f}_i)$, appearing in $\log\mathcal{P}(\mathbb{L}(\phi)^{-1}\tilde{f}_i,\phi|d)$, which is completely invariant to changes in ϕ. This allows our algorithm to make large jumps in ϕ that are completely decoupled from the data and experimental conditions. Notice that this property also extends to posterior sampling and results in fast mixing Gibbs iterations. Indeed, this subtle difference gives a succinct way to see the key advantage gained when working with the lensed parametrization $(\phi,\tilde{f})$ versus unlensed parametrization (ϕ,f).

All of this raises the question: which estimate should one use, $\hat{\phi}_M$ or $\hat{\phi}_J$? Technically, neither $\hat{\phi}_M$ nor $\hat{\phi}_J$ is optimal with respect to the mean squared error (the marginal expected value being the optimal in that case); the remaining suboptimality can reasonably be expected to be small; however, we have not checked this here and leave this to future work. We will see in Sec. V B that there are some apparent advantages to working with $\hat{\phi}_M$ in that the extra determinant term in (11) automatically removes a "mean field" which becomes large in the presence of pixel space masking. On the other hand, if one wishes to sample from the posterior, the extra determinant term in $\hat{\phi}_M$ now becomes a difficult computational obstacle for sampling algorithms. Moreover, the joint $\mathcal{P}(f,\phi|d)$ has the advantage of simultaneously characterizing both the delensed CMB marginal $\mathcal{P}(f|d)$ as well as $\mathcal{P}(\phi|d)$.

III. THE MAXIMIZATION ALGORITHM

With the target probability function (5) in hand, we now describe our maximization algorithm. We have attempted a number of different approaches, but the most efficient we have found is based on the observation that maximizing separately with respect to $\tilde{f}$ and to ϕ cleanly breaks the problem up into two simple pieces, a Wiener filter and something which is independent of the instrument and data. To that end, we employ a coordinate descent, i.e., alternating maximization steps in the $\tilde{f}$ and ϕ directions separately. Coordinate descent also has the advantage that it is essentially the maximization analog to Gibbs sampling, which is exactly the sampling algorithm shown successful for temperature in [13]. We therefore expect the developments that we present here which make the maximization workable for polarization to also transfer to the sampling case.

Consider first the coordinate descent step for $\tilde{f}$. The maximum probability for $\tilde{f}$ given fixed ϕ can be calculated by taking the gradient of the likelihood,

$$\frac{\partial}{\partial\tilde{f}}\log\mathcal{P}(\tilde{f},\phi|d) = (d-\tilde{f})^\dagger\mathbb{C}_n^{-1} - \tilde{f}^\dagger\mathbb{L}(\phi)^{-\dagger}\mathbb{C}_f^{-1}\mathbb{L}(\phi)^{-1}, \quad (16)$$

and setting it to zero. This gives an explicit solution,

$$\tilde{f} = \mathbb{L}(\phi)[\mathbb{C}_f^{-1} + \mathbb{L}(\phi)^\dagger \mathbb{C}_n^{-1}\mathbb{L}(\phi)]^{-1}\mathbb{L}(\phi)^\dagger \mathbb{C}_n^{-1} d, \quad (17)$$

which can be recognized as an ordinary Wiener filtering of the data with a ϕ-dependent signal covariance. The challenge is inverting the quantity in brackets in (17). We find that inverting it with a simple preconditioned conjugate gradient (with a preconditioning matrix that assumes $\phi = 0$ and noise which is diagonal in Fourier space) works sufficiently well. The reduction of part of the problem to the well known Wiener filter problem is a major advantage of the coordinate descent, since many Wiener filter algorithms exist which are efficient and can be guaranteed to converge, unlike generic nonlinear optimization algorithms.

Now consider the coordinate descent step for ϕ. Here the gradient is given by,

$$\begin{aligned}\frac{\partial}{\partial\phi}\log\mathcal{P}(\tilde{f},\phi|d) &= -\frac{1}{2}\frac{\partial}{\partial\phi}[\tilde{f}^\dagger\mathbb{L}(\phi)^{-\dagger}\mathbb{C}_f^{-1}\mathbb{L}(\phi)^{-1}\tilde{f}] - \phi^\dagger\mathbb{C}_\phi^{-1} \\ &= -\tilde{f}^\dagger\mathbb{L}(\phi)^{-\dagger}\mathbb{C}_f^{-1}[\frac{\partial}{\partial\phi}\mathbb{L}(\phi)^{-1}\tilde{f}] - \phi^\dagger\mathbb{C}_\phi^{-1}. \quad (18)\end{aligned}$$

Taking the adjoint and setting to zero yields

$$\left[\frac{\partial}{\partial\phi}\mathbb{L}(\phi)^{-1}\tilde{f}\right]^\dagger \mathbb{C}_f^{-1}\mathbb{L}(\phi)^{-1}\tilde{f} - \mathbb{C}_\phi^{-1}\phi = 0. \quad (19)$$

Unlike the $\tilde{f}$ step, it is not possible to obtain an explicit solution for ϕ. Instead, we solve this iteratively with a quasi Newton-Raphson step,

$$\phi_{i+1} = \phi_i - \alpha\mathbb{H}(\tilde{f},\phi_i)^{-1}\frac{\partial}{\partial\phi}\log\mathcal{P}(\tilde{f},\phi_i|d). \quad (20)$$

Here $\mathbb{H}(\tilde{f},\phi_i)$ denotes the Hessian of $\phi \mapsto \mathcal{P}(\tilde{f},\phi|d)$ and α is a scalar coefficient over which we perform a line search to maximize the probability. We take $\mathbb{H} \approx \mathbb{C}_\phi$, which is the contribution to the Hessian from only the ϕ-prior term, but which we find works extremely well in practice. By the time we are close to maximum, we expect a single Newton-Raphson step would take us quite close to the exact solution of (19), but we have found that even before we reach the maximum we can get away with just a single iteration of (20) at each coordinate descent step and convergence is still quite fast.

For the ϕ step, the coordinate descent has removed all explicit dependence on the instrument; note that neither the data nor the noise covariance (and hence no masking, transfer function, etc.) appear explicitly in (19). It is worth restating that this would *not* have been the case if we were performing coordinate descent with respect to (f,ϕ) as opposed to $(\tilde{f},\phi)$; hence this can be seen as another fundamental advantage of the lensed parametrization.

The maximization algorithm then simply starts at $\phi = 0$ and alternates these two coordinate descent steps, until acceptable convergence is reached. There is only one additional detail we need to describe which is necessary for convergence to happen efficiently enough, and that is our use of a cooling schedule for the covariance, $\mathbb{C}_f$. By this we mean that we replace $\mathbb{C}_f$ everywhere that it appears in the iterating equations with a new covariance, which we call the cooling covariance and denote with $\hat{\mathbb{C}}_f$. It is initially set to the *lensed* CMB covariance (which we will denote by $\tilde{\mathbb{C}}_f$), then progressively "cooled" it towards $\mathbb{C}_f$. By the final iteration we cool to exactly $\mathbb{C}_f$ and thus are maximizing the true posterior.

The cooling scheme is aimed at keeping the power spectrum of the $\tilde{f}$ estimate constant across iterations and roughly matching the expected power spectrum of the lensed CMB. This happens at the expense of making the power spectrum of f not always match the unlensed spectrum, but is advantageous nevertheless since it is in the lensed parametrization that we are performing the coordinate descent. To achieve this goal, the cooling scheme takes $\hat{\mathbb{C}}_f$ at a given iteration to be the expected power spectrum of the true lensed field delensed by the current ϕ estimate at that iteration. For a given configuration (i.e., noise level, pixelization, map size, etc.), we can calculate this covariance with simulations, since we have access to the true lensed field. In fact, we find that only one simulation is necessary, as we can greatly reduce sample variance fluctuations by modeling the cooling covariance as a geometric mean between the lensed and unlensed C_ℓ's, with an ℓ-dependent weight, w_ℓ, and heavily interpolating this quantity based on the observed BB spectrum of the one simulation. This produces a set of geometric weights w_ℓ^i for each iteration i which we use in subsequent runs. These weights, along with the data and the number of iterations are the only inputs to the maximization procedure, which we summarize in Algorithm 1 below.

Algorithm 1. Joint posterior maximization.

procedure JOINTPOSTERIORMAX(d, N, w_ℓ^i)
  $\phi_1 = 0$, $f_1 = 0$, $\tilde{f}_1 = 0$
  for $i = 1...N-1$ **do**
    $\hat{\mathbb{C}}_{f,\ell} = (\mathbb{C}_{f,\ell})^{w_\ell^i}(\tilde{\mathbb{C}}_{f,\ell})^{1-w_\ell^i}$
    $\mathbb{A} = \hat{\mathbb{C}}_f^{-1} + \mathbb{L}(\phi_i)^\dagger\mathbb{C}_n^{-1}\mathbb{L}(\phi_i)$
    $b = \mathbb{L}(\phi_i)^\dagger\mathbb{C}_n^{-1}d$
    $f_{i+1} = \mathbb{A}^{-1}b$ ▷ Solve via CG
    $\tilde{f}_{i+1} = \mathbb{L}(\phi_i)f_{i+1}$
    $g = [\frac{\partial}{\partial\phi}\mathbb{L}(\phi_i)^{-1}\tilde{f}_{i+1}]^\dagger\hat{\mathbb{C}}_f^{-1}f_{i+1} + \mathbb{C}_\phi^{-1}\phi_i$
    $\alpha = \mathsf{Max}_\alpha\mathcal{P}(\tilde{f}_i, \phi_i - \alpha\mathbb{C}_\phi g|d)$
    $\phi_{i+1} = \phi_i - \alpha\mathbb{C}_\phi g$
  end For
  return $\phi_N, f_N, \tilde{f}_N$
end procedure

We have already ascertained in the previous section that the lensing operation which appears throughout the algorithm, or more specifically its inverse, needs to be area-preserving. Thus a requirement on the lensing algorithm which we use is,

(1) $|\det(\mathbb{L}(\phi)^{-1})| = 1$ to numerical precision.

Examining Algorithm 1, we note that we also need two other things of the lensing operation,

(2) Computation of $\mathbb{L}(\phi)^{\dagger} f$

(3) Computation of $[\frac{\partial}{\partial\phi}\mathbb{L}(\phi)^{-1}\tilde{f}]^{\dagger}$.

In the next section we develop LenseFlow which performs pixelized lensing in a way that simultaneously satisfies (1), (2), and (3) above.

IV. LENSEFLOW

LenseFlow is an algorithm that utilizes an ordinary different equation (ODE) to describe the lensing operator, $\mathbb{L}(\phi)$. An auxiliary "time" variable is introduced which continuously connects the lensed and unlensed maps such that $\mathbb{L}(\phi)f$ is given by the solution of an ODE over map pixels with initial conditions f. Because the ODE is homogenous, we can regard the pixel values as "flowing" from their unlensed values to their lensed ones, hence the name LenseFlow. There are a number of advantages one obtains with an ODE characterization of a linear operator. First, operator inversion simply corresponds to running the ODE in reverse. Secondly, log determinants can be analyzed using the trace of the velocity operator, integrated over time. Finally, in many cases higher order derivatives with respect to both initial conditions and parameters of the ODE have their own ODE characterizations. In the case of LenseFlow, these enable fast and accurate calculation of gradient and Hessian operators of $\log \mathcal{P}(\tilde{f}, \phi|d)$ with respect to both $\tilde{f}$ and ϕ.[3]

We begin to define LenseFlow by introducing an artificial time variable to the CMB field which connects the lensed CMB at $t = 1$ with the unlensed CMB at $t = 0$. In particular, for $t \in [0, 1]$ let

$$f_t(x) \equiv f(x + t\nabla\phi(x)) \tag{21}$$

so that $f_0(x) = f(x)$ and $f_1(x) = \tilde{f}(x)$. An ordinary differential equation for f_t can be derived from

$$\frac{df_t(x)}{dt} = \nabla^i f(x + t\nabla\phi(x))[\nabla\phi(x)]^i, \tag{22}$$

and the following chain rule

$$\nabla^i f_t(x) = \nabla^j f(x + t\nabla\phi(x))[\delta^{ij} + t\nabla^i\nabla^j\phi(x)], \tag{23}$$

where $\nabla^i \equiv \partial/\partial x^i$ (we are working here in the flat-sky approximation) and δ^{ij} is the Kronecker delta. The quantity in brackets in (23) represents the 2×2 Jacobian of the map $x \mapsto x + t\nabla\phi(x)$, which for $t = 1$ is often called the magnification matrix; we will henceforth label it with $\mathbb{M}_t$. It is invertible in the weak lensing regime in which we work here; thus we can combine the above two equations to yield that f_t satisfies

$$\dot{f}_t = (\nabla^j\phi)(\mathbb{M}_t^{-1})^{ji}\nabla^i f_t. \tag{24}$$

By definition, solving the ODE (24) forward in time, $t = 0 \to 1$, represents the lensing operation. Moreover, exact inverse lensing simply corresponds to flowing the ODE backwards in time, $t = 1 \to 0$. Notice that invertibility of LenseFlow also extends to discrete pixel-to-pixel lensing by replacing the gradient, ∇, in (24), with its discrete Fourier analog.

The fact that LenseFlow is an area preserving linear operator, i.e., that (1) holds, follows directly from (24). To see why, first define

$$p_t^i = (\nabla^j\phi)(\mathbb{M}_t^{-1})^{ji} \tag{25}$$

so that (24) is written in compact form $\dot{f}_t = p_t^i\nabla^i f_t$. Now since the flow from f_0 to f_1 can be written as composition of infinitesimally small linear operations, the lensing operator $\mathbb{L}(\phi)$ is decomposed as follows:

$$f_1 = \underbrace{[1 + \epsilon p_{t_n}^i\nabla^i]\cdots[1 + \epsilon p_{t_0}^i\nabla^i]}_{=\mathbb{L}(\phi)} f_0, \tag{26}$$

where $\epsilon = \frac{1}{n} = t_{i+1} - t_i$ and $t_0 = 0$. Notice that

$$\begin{aligned}\log\det[1 + \epsilon p_t^i\nabla^i] &= \epsilon\mathrm{Tr}[p_t^i\nabla^i] + \mathcal{O}(\epsilon^2)\\ &= \mathcal{O}(\epsilon^2),\end{aligned} \tag{27}$$

where the last equality follows since the operator ∇^i is Hermitian antisymmetric. This applies also to the inverse operation, thus up to ODE time-step discretization error, condition (1) holds for LenseFlow, independent of pixel size,

$$\lim_{\epsilon\to 0}\det(\mathbb{L}(\phi)^{-1}) = 1. \tag{28}$$

In Sec. VII, we verify this numerically.

It will be useful to have a compact notation for the decomposition of a linear operator characterized by an ODE, as in (26). To that end define

$$\underset{t=t_0\to t_n}{\mathsf{ODE}}\{\mathbb{V}_t\} \equiv [1 + \epsilon\mathbb{V}_{t_n}]\cdots[1 + \epsilon\mathbb{V}_{t_0}], \tag{29}$$

where $\mathbb{V}_t$ represents a "velocity operator" generating an ODE of the form $\dot{f}_t = \mathbb{V}_t f_t$ and where $\epsilon = t_{i+1} - t_i$

[3]Incidentally, the ODEs for calculating these derivatives are exactly analogous to the backpropagation techniques used for learning deep neural networks [17] but are derived here completely from ODE theory.

represents an infinitesimal time step for an ordered equidistant sequence of time points $t_0, t_1, \ldots, t_n$. This allows us to succinctly define LenseFlow as,

$$\mathbb{L}(\phi) = \underset{t=0\to 1}{\mathsf{ODE}}\{p_t^i \nabla^i\}. \tag{30}$$

The infinitesimal ODE expansion also makes it clear that both the inverse and adjoint of an ODE operator is also an ODE operator, but with time reversed, and in the latter case with a negative adjoint velocity,

$$[\underset{t=t_0\to t_n}{\mathsf{ODE}}\{\mathbb{V}_t\}]^{-1} = \underset{t=t_n\to t_0}{\mathsf{ODE}}\{\mathbb{V}_t\} \tag{31}$$

$$[\underset{t=t_0\to t_n}{\mathsf{ODE}}\{\mathbb{V}_t\}]^{\dagger} = \underset{t=t_n\to t_0}{\mathsf{ODE}}\{-\mathbb{V}_t^{\dagger}\}. \tag{32}$$

Due to the fact that $[p_t \cdot \nabla]^{\dagger} f = -\nabla^i(p_t^i f)$, the latter equation can be used to compute the adjoint lensing operator

$$\mathbb{L}(\phi)^{\dagger} = \underset{t=1\to 0}{\mathsf{ODE}}\{\nabla^i(p_t^i \bullet)\}, \tag{33}$$

where the expression $\nabla^i(p_t^i \bullet)$ is shorthand for the operator $f \mapsto \nabla^i(p_t^i f)$. Notice that (33) achieves (2), another of our requirements for the lensing operation. Although not explicitly needed, note also that the operator $\mathbb{L}(\phi)^{-\dagger}$ is conveniently computed by simply applying a time reversal of (33), as per (31).

For the final requirement in (3), we need to compute derivatives of the inverse lensing operator with respect ϕ and initial condition, f_0. Introducing infinitesimal perturbations $\delta\phi$ and δf_t into (24), we have

$$\begin{aligned}\dot{\delta f}_t = {} & (\nabla^i \delta\phi)(\mathbb{M}_t^{-1})^{ij}\nabla^j f_t + (\nabla^i\phi)\delta(\mathbb{M}_t^{-1})^{ij}\nabla^j f_t \\ & + (\nabla^i\phi)(\mathbb{M}_t^{-1})^{ij}\nabla^j \delta f_t.\end{aligned} \tag{34}$$

Simplifying $\delta(\mathbb{M}_t^{-1})^{ij}$ and treating $\delta\phi$ as a time dependent variable results in

$$\begin{bmatrix} \dot{\delta f}_t \\ \dot{\delta\phi}_t \end{bmatrix} = \begin{bmatrix} p_t^i\nabla^i & v_t^i\nabla^i - t\mathbb{W}_t^{ij}\nabla^i\nabla^j \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \delta f_t \\ \delta\phi_t \end{bmatrix}, \tag{35}$$

where p_t, v_t, and $\mathbb{W}_t$ are defined by

$$p_t^i = (\nabla^j\phi)(\mathbb{M}_t^{-1})^{ji} \tag{36}$$

$$v_t^i = (\nabla^j f_t)(\mathbb{M}_t^{-1})^{ji} \tag{37}$$

$$\mathbb{W}_t^{ij} = (\nabla^p\phi)(\nabla^q f_t)(\mathbb{M}_t^{-1})^{pi}(\mathbb{M}_t^{-1})^{jq} \tag{38}$$

(the definition of p_t is repeated here for clarity). It is important to note that, unlike p_t^i which is a scalar field for each index i, the quantities $\mathbb{W}_t^{ij}$ and v_t^i are instead a TQU vector of temperature and polarization fields at each index. As is usually implicitly assumed, multiplication between a scalar field and a TQU vector broadcasts over the TQU indices. One important consequence of this is that the adjoint of $\mathbb{W}_t^{ij}\nabla^i\nabla^j$ and $v_t^i\nabla^i$ are given by $\nabla^j\nabla^i((\mathbb{W}_t^{ij})^{\mathsf{T}}\bullet)$ and $-\nabla^i((v_t^i)^{\mathsf{T}}\bullet)$, respectively, where we define T to represent a transposition of just the TQU indices. For example, if f is a TQU vector of fields, $f^{\mathsf{T}}f$ represents the scalar field $I^2 + Q^2 + U^2$ (in contrast to $f^{\dagger}f$, for example, which would be a single number).

If we now consider a map between the lensed and unlensed parametrizations, $(f, \phi) \mapsto (\tilde{f}, \phi)$, the Jacobian $\mathbb{J} \equiv \frac{\partial(\tilde{f},\phi)}{\partial(f,\phi)}$ and its inverse are given by

$$\mathbb{J} = \begin{bmatrix} \frac{\partial\tilde{f}}{\partial f} & \frac{\partial\tilde{f}}{\partial\phi} \\ 0 & 1 \end{bmatrix}\mathbb{J}^{-1} = \begin{bmatrix} \frac{\partial f}{\partial\tilde{f}} & \frac{\partial f}{\partial\phi} \\ 0 & 1 \end{bmatrix}. \tag{39}$$

Equations (35)–(38) show that $\mathbb{J}$ can be computed as

$$\mathbb{J} = \underset{t=0\to 1}{\mathsf{ODE}}\left\{\begin{bmatrix} p_t^i\nabla^i & v_t^i\nabla^i - t\mathbb{W}_t^{ij}\nabla^i\nabla^j \\ 0 & 0 \end{bmatrix}\right\}, \tag{40}$$

and (32) immediately gives that the adjoint Jacobian is

$$\mathbb{J}^{\dagger} = \underset{t=1\to 0}{\mathsf{ODE}}\left\{\begin{bmatrix} \nabla^i(p_t^i\bullet) & 0 \\ \nabla^i((v_t^i)^{\mathsf{T}}\bullet) + t\nabla^j\nabla^i((\mathbb{W}_t^{ij})^{\mathsf{T}}\bullet) & 0 \end{bmatrix}\right\}. \tag{41}$$

Note that the velocities for the Jacobian ODE depend on f_t, which can be precomputed from an initial application of the corresponding lensing operator, or in some cases simply solved for in unison.

As before, the inverse of (41) can be trivially computed by time reversal of the ODE, using (31). The bottom left block of $\mathbb{J}^{-\dagger}$ then satisfies

$$\mathbb{J}^{-\dagger}\begin{bmatrix} \delta f \\ 0 \end{bmatrix} = \begin{bmatrix} * \\ \left[\frac{\partial}{\partial\phi}\mathbb{L}(\phi)^{-1}\tilde{f}\right]^{\dagger}\delta f \end{bmatrix},$$

which is exactly the necessary derivative which satisfies the final requirement of (3).

Although Hessians are not needed for our iterating equations, we remark that by a process analogous to inserting infinitesimal perturbations to (34), one can create an ODE flow for the lensing Hessian starting from the Jacobian ODE. This Hessian operator cannot be stored in practice for realistically sized maps, but can be applied in the same computational order as the lensing and Jacobian operations themselves. This could prove very useful for sampling algorithms, for example aiding in computing the mass matrix in a Hamiltonian Monte-Carlo sampler.

V. RESULTS

We now begin to test our algorithm on simulations. The generic form of the simulated data presented in this section have the form

$$d = \mathbb{M}\mathbb{L}(\phi)f + n \tag{42}$$

$$= \mathbb{M}\tilde{f} + n, \tag{43}$$

where $\mathbb{M}$ is an operator that incorporates a beam, a pixel mask (which zeroes out certain pixels) and a frequency cut (which zeros out frequencies above a specified $\ell_{\rm max}$). In the previous sections, $\mathbb{M}$ was omitted in the interest of simplifying the exposition. Incorporating $\mathbb{M}$ modifies the posterior, $\mathcal{P}(\tilde{f}, \phi|d)$, only slightly, giving

$$\begin{aligned} -2\log\mathcal{P}(\tilde{f},\phi|d) = {} & (d-\mathbb{M}\tilde{f})^\dagger \mathbb{C}_n^{-1}(d-\mathbb{M}\tilde{f}) \\ & + \tilde{f}^\dagger \mathbb{L}(\phi)^{-\dagger}\mathbb{C}_f^{-1}\mathbb{L}(\phi)^{-1}\tilde{f} \\ & + \phi^\dagger \mathbb{C}_\phi^{-1}\phi. \end{aligned} \tag{44}$$

In terms of the joint posterior maximization, the only adjustment is the definition of variables A and b given in steps 5 and 6 in Algorithm 1, which become[4]

$$A = \hat{\mathbb{C}}_f^{-1} + \mathbb{L}(\phi_i)^\dagger \mathbb{M}^\dagger \mathbb{C}_n^{-1}\mathbb{M}\mathbb{L}(\phi_i) \tag{45}$$

$$b = \mathbb{L}(\phi_i)^\dagger \mathbb{M}^\dagger \mathbb{C}_n^{-1} d. \tag{46}$$

We generate simulated data with CMB-S4 like noise properties, since it is for these low noise levels that one expects to see a major benefit of the optimal procedure. We assume 1 μK-arcmin Gaussian temperature noise, scaled by $\sqrt{2}$ for polarization, and a 3 arcmin Gaussian beam [18]. Additionally, at low multipoles we adjust the noise power spectrum to mimic a $1/f$ knee. Specifically, we take $\ell_{\rm knee} = 100$ and $\alpha_{\rm knee} = 3$ according to the parametrization of [19], who suggest that for a large aperture array this would be the maximum allowable knee frequency to be competitive with other configurations. This, in effect, lets us test the maximal but realistic impact of a nonwhite noise power spectrum on our procedure.

We use pixels which are 3 arcmin on a side, which are fairly large compared to typical analyses. This highlights one of the advantages of LenseFlow, which is that we get numerically stable and accurate lensing with determinant equal to exactly unity even on such large pixels. At fixed map size, this makes the algorithm faster because of the smaller matrix operations involved. The runs described here use maps which are 512×512 pixels, which at this resolution correspond to around 600 deg^2, comparable to currently existing polarization datasets to which our procedure would be naturally applicable (e.g., [20,21]). The Nyquist frequency for 3 arcmin pixels is $\ell = 3600$, above which we expect little cosmological information in our setup. Nevertheless, we have also verified the algorithm with 1 arcmin pixels, and find the main difference is just a longer time-to-convergence for the conjugate gradient.

We generate a Gaussian random realization of the CMB from a fiducial CMB spectrum with cosmological parameters given by their posterior mean given the *Planck* 2015 TT data [22], combined with the updated *Planck* high frequency instrument large scale polarization data τ [23]. We take $r_{0.002} = 0.05$, compatible with current upper bounds [24].

A. Without map-level masking

Using the configuration just described, we create one main simulated dataset. The resulting temperature and polarization maps are shown in Fig. 4. Note that although this figure shows a pixel mask, in this first section we consider only Fourier-space masking (we will add map-level masking in Sec. V B). The Fourier mask we use in this section is an unapodized low-pass filter at $\ell = 3000$.

We run 50 iterations of the algorithm on this data, the entire run completing in around two hours on a single Intel Haswell 2.3 GHz 16-core CPU.[5] In Fig. 1 we see the excellent visual agreement between the true ϕ and lensed and unlensed B maps and the ones recovered by the algorithm. We expect these should resemble something like a Wiener filter solution, and thus have low signal-to-noise modes attenuated; the signal-to-noise is low enough that this is visually apparent only for the unlensed B map. Figure 2 shows the power spectrum of these maps, where one can see the attenuation for all cases, as well as the very small residual at medium and large scales between the reconstructed ϕ map and the truth.

These maps and power spectra look as one might expect for a MAP estimate, but we would like a more robust way

[4]Notice that (42) implies that masking is absent from the noise realization, n. The main reason our simulations are configured this way is to ensure that the operator $\mathbb{C}_n$ which appears in (45)–(46) is nonsingular and easily invertible even in the presence of pixel and Fourier masking. In the case of simulated data, it is trivial to leave the noise realization unmasked. For real data, one can fill in masked regions with a simulated realization from the noise model, which leaves the statistical properties of the noise unchanged in the case of perfectly white noise. Although the assumption of white noise is never exact in real data, we believe the technique of artificially adding noise to masked pixels has the potential to still yield accurate Wiener filter approximations in the presence of complicated masking even for more realistic experimental noise models.

[5]As the algorithm itself is entirely sequential, no parallelization is employed aside from using a multithreaded fast fourier transform library and making use of single instruction multiple data vectorization for point-wise matrix multiplications. The run-time is dominated by computing the LenseFlow ODE velocity during the Runge-Kutta integration for the lensing operations performed in the CG step. The asymptotic complexity is set by the fast fourier transform and is thus $O(N \log N)$ where N is the number of pixels in the map, although in practice we find speed difference between lensing; e.g., a 1024×1024 and 2048×2048 map is a bit worse than this because the bottleneck is memory access.

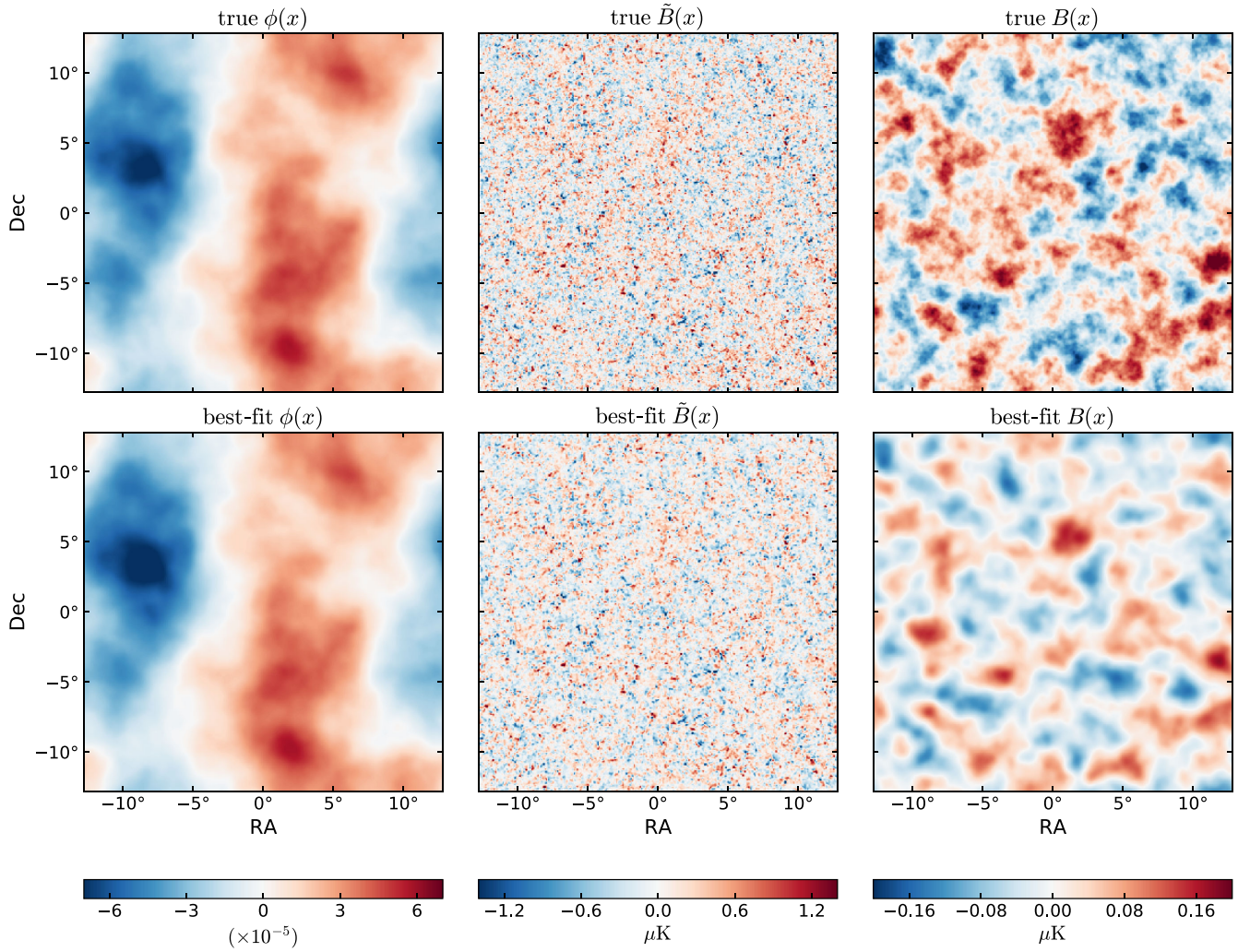

FIG. 1. The reconstructed ϕ and lensed/unlensed B maps from a run of our algorithm on simulated data (bottom row), as compared to the simulation truth (top row). This is for the run with only Fourier-space masking described in Sec. V A. The reconstruction, as expected, resembles a Wiener filter solution wherein low signal-to-noise modes are attenuated.

to verify that we have attained the true maximum. One way to do so is to compute the χ^2 expected at the best-fit point and compare to what we actually achieved. By χ^2, we are referring to the sum of the terms in (3) excluding the determinants, i.e., the sum of the χ^2's of the data residual, f, and ϕ, with respect to $\mathbb{C}_n$, $\mathbb{C}_f$, and $\mathbb{C}_\phi$, respectively. Approximating the problem as linear, we expect the best-fit χ^2 to scatter according to a χ^2 distribution with degrees of freedom given by the total number of unmasked pixels in the three terms, minus the number of free parameters which are fit for. In Fig. 3 we show the one, two, and three sigma regions for this expectation as the gray bands. The χ^2 after each of the 50 steps of the algorithm is also plotted, both with respect to the true covariance, $\mathbb{C}_f$, and with respect to the cooling covariance, $\hat{\mathbb{C}}_f$. By the final iteration when we fully cool the covariance, we are well within this gray band, a good indication of convergence.

Although this result is suggestive that we have successfully converged, our problem is not exactly linear, so we cannot rule out that the true expected distribution of best-fit χ^2 is actually lower. Another test we can perform is to examine the gradient of the posterior after each iteration. As we reach a local or global maximum, we expect the gradient to approach zero. Since the gradient in the $\tilde{f}$ direction is always reduced to zero up to numerical precision by the Wiener filter step, we examine the gradient in the ϕ direction. Here, we find that across all scales, the power spectrum of the gradient drops by several orders of magnitude during the 50 iterations of the algorithm, until hitting a numerical floor. Taken together, that the best-fit maps and power spectrum look as expected given the simulation ground truth, that we are close to the expected χ^2, and that the gradient is approaching zero are strong indications that the algorithm has reached the global maximum.

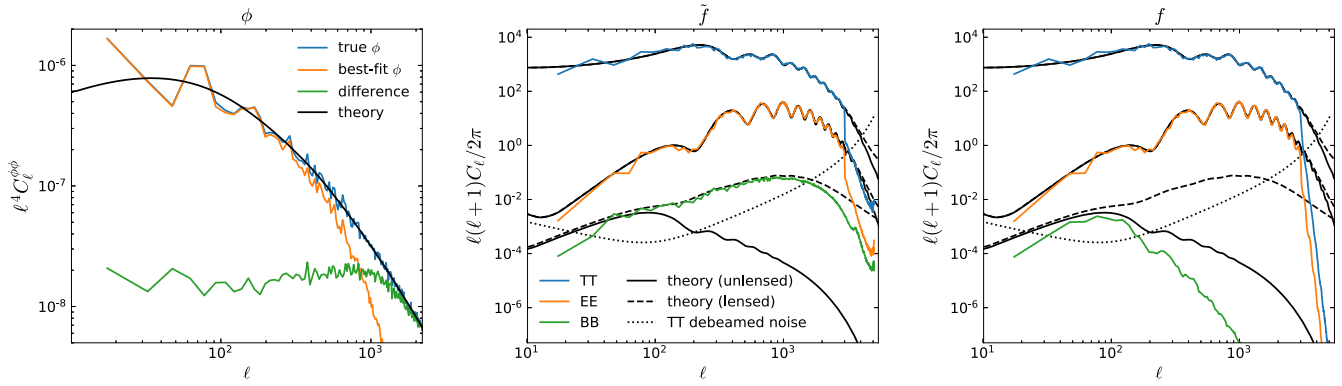

FIG. 2. The power spectra of the best-fit ϕ and lensed/unlensed CMB maps from a run of our algorithm, as compared to the input theory spectra. This is for the run with only Fourier-space masking described in Sec. V A (the same run for which maps are shown in Fig. 1). The left panel also shows the power spectrum of the simulation truth for the ϕ map itself as well as the power spectrum of the difference between this and our reconstructed solution, demonstrating the fidelity of the reconstruction. The "bump" visible in the lensed spectra near the Nyquist frequency at $\ell = 3600$ signals the smallest scale for which the LenseFlow pixelized lensing approximation is accurate at this pixel size (similar features are produced by other lensing algorithms). We mask the data in Fourier space beyond $\ell = 3000$ so that we are not sensitive to this region, and the effects of this mask are visible above as a sharp suppression in power at $\ell > 3000$. **Note:** The beam-deconvolved noise spectrum shown in the middle and right panel are used to illustrate the effective signal-to-noise ratio in the simulated data. The actual simulation data used to generate the best-fit ϕ and lensed/unlensed CMB maps are not beam-deconvolved. See Sec. V for details.

B. With map-level masking

We now turn to demonstrating that the algorithm works when we apply map-level masking. Such masking is necessary in any real analysis as various sources of galactic and extragalactic contamination are most efficiently dealt with by directly excising them from the maps. Here we randomly place 100 point sources holes with radii between 5 and 10 arcmin. Additionally, for a flat-sky analysis as performed here, it is necessary to include a border mask so as to "embed" the observed sky patch (which is nonperiodic) onto a Fourier grid with is otherwise assumed periodic. To this end, we apply a 2° border mask. Both the border mask and the point source mask are mildly apodized.

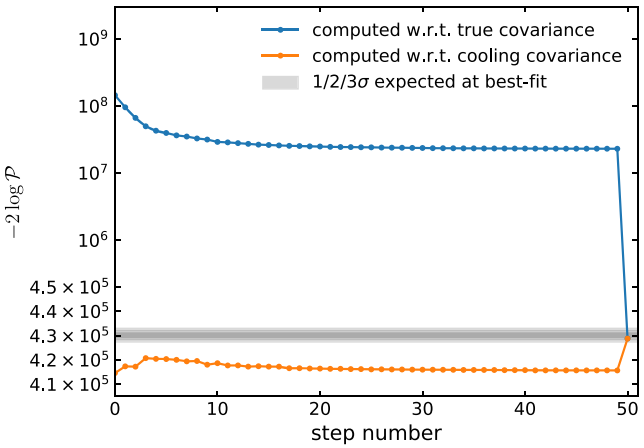

FIG. 3. The posterior probability after each iteration of our algorithm during the run on the simulated dataset described in Sec. V. The top (blue) line is the posterior with respect to the true covariance, and the bottom (orange) line is with respect to the cooling covariance (note the y-scale is mixed log and linear). For the final step these two are identical since the cooling covariance is fully cooled and equals the true covariance. The grey band represents the value of the posterior probability expected at the best-fit point, and our best fit sits well within this expectation.

We use the identical simulated data shown in Fig. 4 as in the previous section, with the only change being that we apply this map-level mask. Note that we continue to apply the Fourier mask which removes $\ell > 3000$, hence here we are testing the performance of the algorithm in the presence of masking which is not diagonal in either map or Fourier space. This introduces a subtle nontriviality in inverting the noise covariance of the masked data, which we account for here with a trick of filling in the masked regions of the map with a realization of noise from $\mathbb{C}_n$. The data, as well as the mask, is shown in Fig. 4.

Two small changes to the algorithm itself are necessary as compared to the unmasked run. First, the cooling weights are recomputed for the specific mask, although using the same procedure as described earlier. Second, not surprisingly, the Wiener filter requires more steps to achieve satisfactory accuracy.[6] That no other major changes to the algorithm are required might have been expected because, as mentioned earlier, one fundamentally nice

[6]In fact, to ease convergence in some cases we find it necessary to replace the one-dimensional line-search $\phi_i - \alpha\mathbb{C}_\phi g$ over α with a two dimensional line-search $\phi_i - \alpha_1\mathbb{C}_\phi g - \alpha_2\psi$ over (α_1, α_2) where ψ is defined as the inverse Laplacian of the border mask and is designed to approximate the mean-field feature described later in Sec. V B. This modification appears to improve numerical stability in Algorithm 1, but is not necessary in all configurations we have tried, so we mention it here but do not discuss it further.

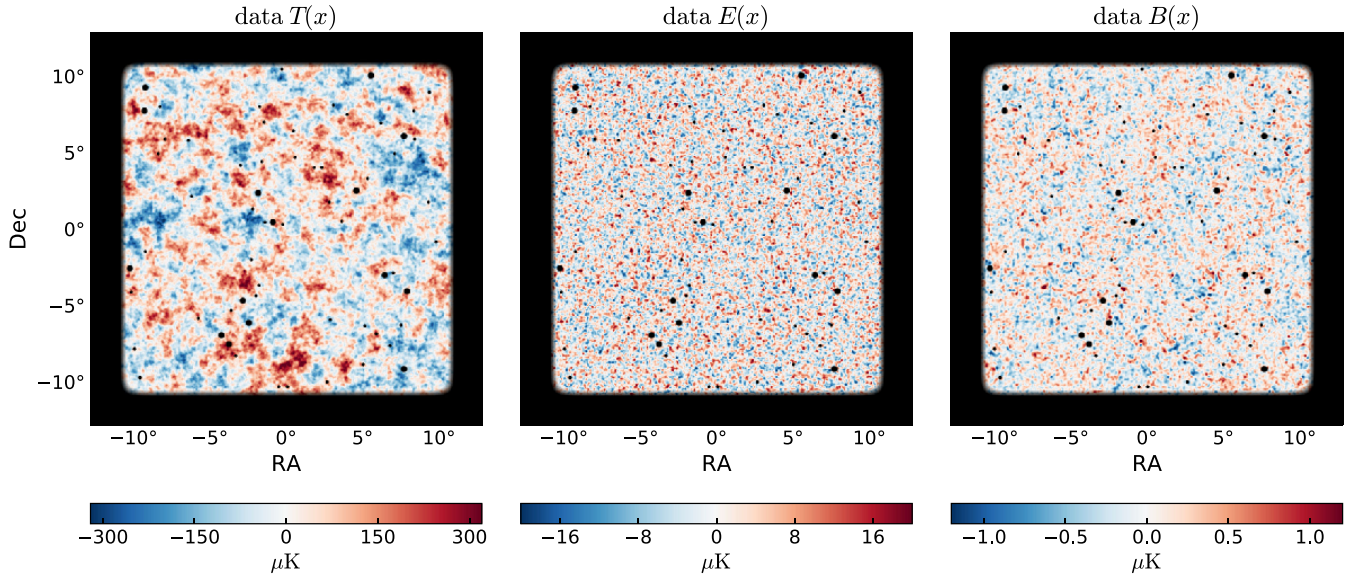

FIG. 4. The simulated data used in the runs described in Sec. V. We use a 512×512 grid with 3 arcmin pixels, which covers roughly 600 deg^2. It assumes a setup approximating an expected CMB-S4 configuration, with a 3 arcmin beam and stationary 1 μK-arcmin temperature noise, modulated to include a $1/f$ contribution below $\ell_{\rm knee} = 100$ (see text for more details). One hundred unapodized point sources with radii between 5 and 10 arcmin are randomly placed within the region. A 2° mildly apodized border mask is applied, as well as a Fourier-space cut above $\ell > 3000$. Note that for this figure the mask is simply overlayed on the unmasked T, E, and B images rather being multiplied into T, Q, and U as is done in the likelihood, since multiplying it in would result in large E to B leakage spoiling the ability to see B. Additionally, the unmasked data has been Wiener filtered with the lensed CMB covariance as the signal covariance to reduce the visual impact of noise.

feature of the lensed parametrization is that it removes from the ϕ step any explicit dependence on the instrument or dataset (i.e., on masking). Of course, there could have been an impact on the decorrelating effect of switching to the lensed parametrization itself, or on the effectiveness of the quasi Newton-Raphson step, but neither appears to be the case. This is good news as it means that if one wishes to even further improve the performance of the algorithm, one needs to focus only on improving the Wiener filter, where many more sophisticated methods exist other than the fairly rudimentary preconditioned conjugate gradient which we have found sufficient here [e.g., [25–29]].

Figure 5 shows the unlensed CMB estimate $\hat{f}_J$ compared the simulation truth. We find, as expected, a Wiener filterlike solution with low signal-to-noise modes attenuated as is visible for B, and with power slowly decaying towards zero in the masked regions as is visible for T, E, and B.

The lensing potential estimate $\hat{\phi}_J$ corresponding to $\hat{f}_J$ is shown in Fig. 6 (bottom left). Notice what appears to be a large scale "bias" in the estimate $\hat{\phi}_J$ as compared to the true ϕ (top left). This feature corresponds to a so called "mean field", akin to the one which must be subtracted to debias the quadratic estimator. Similarly as for the quadratic estimate, it arises because the mask induces correlations between different ℓ-modes, which the best fit then attributes to lensing. We remark that the marginal estimate $\hat{\phi}_M$ would not show this feature because it is implicitly corrected for by the determinant term found in the marginalized posterior (11) which is not present in the joint posterior (3).

The effect of the mean field bias in $\hat{\phi}_J$ is simpler when considering the convergence $\kappa \equiv -\nabla^2\phi/2$. There, the mean field roughly translates to an additive constant offset over nonmasked pixels,

$$\hat{\kappa}_J(x) \approx \mu + \kappa(x) \quad \text{for all nonmasked pixels } x. \tag{47}$$

Intuitively this can be understood as follows. Because in the masked regions the Wiener filterlike suppression drives the solution to zero, in the absence of lensing this leads to an f power spectrum which, on average across the entire map, is smaller than expected given $\mathcal{C}_f$. Now note that since the CMB has a mostly "red" spectrum (i.e., tilted to the right), an overall magnification has a similar effect to reducing the overall amplitude.[7] Thus with the lensing potential available as a free parameter, the best fit is able to slightly increase f to better agree with its covariance, but add an overall magnification to ϕ so that $\tilde{f}$ is reduced and still agrees with the data.

This effect can be seen in the middle column of Fig. 6 where the fluctuations of $\hat{\kappa}_J(x)$ (bottom middle) track the true $\kappa(x)$ (top middle, plotted with an additional beam to make the relevant scales more visible). Notice that the

[7]This degeneracy is in fact exact for power-law spectra in the limit of infinite-size maps [30].

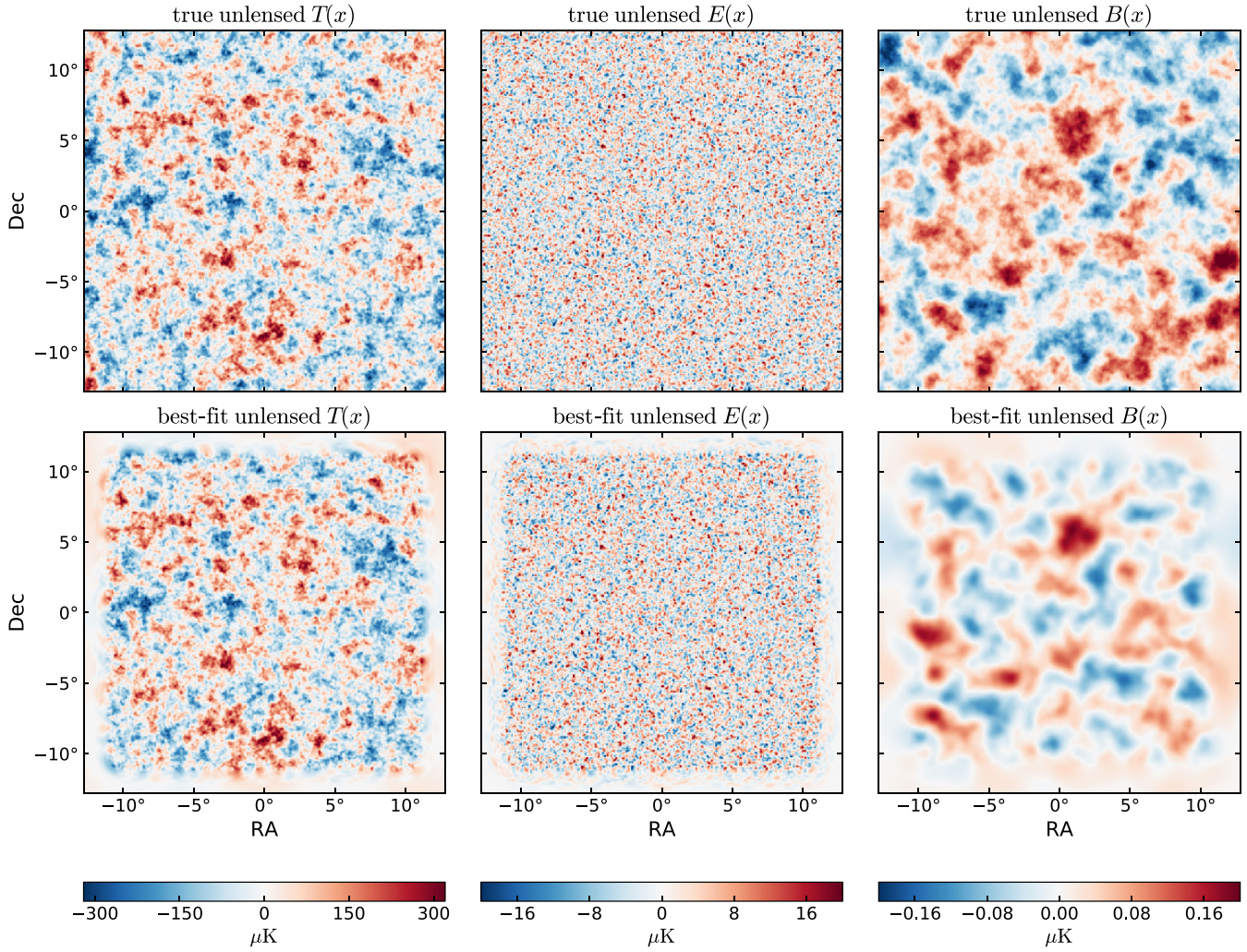

FIG. 5. The reconstructed unlensed T, E, and B maps from a run of our algorithm on simulated data (bottom row), as compared to the simulation truth (top row). This is for the run discussed in Sec. V B which includes the real-space mask that is visible in Fig. 4. As expected, low signal-to-noise modes are attenuated and the solution provides a partial reconstruction even in the masked region.

average value of $\hat{\kappa}_J(x)$ over nonmasked pixels appears slightly smaller than zero. This is the mean field and results in a more visually dramatic effect on the original non-Laplacian scale (as seen in the bottom left image). To probe the accuracy of the smaller scale fluctuations one can recenter $\hat{\kappa}_J$ and κ to have zero mean over nonmasked pixels, then set any masked pixels to zero so that only errors within the observation region are probed. The resulting error bandpowers are shown in Fig. 7 and can be seen to be similar to what one expects from nonmasked observations. Applying $-2\nabla^{-2}$ to the recentered and mask-attenuated $\hat{\kappa}_J$, which we refer to as "deprojecting" in the figure captions, has the effect of visually removing the mean field features in the original estimate (shown bottom right in Fig. 6 with the corresponding operation applied to the true ϕ shown top right).

As in the previous section, we would like to confirm convergence, thus ascertaining that the mean field is a real feature of the global MAP estimate and not a local mode or artifact of Algorithm 1. The first piece of evidence is that the best fit, similarly as before, attains an acceptable best-fit χ^2, in this case 0.8σ above expectation. Going beyond just this one simulated dataset, we also check the distribution of best-fit χ^2's on 100 other simulations (with somewhat smaller map sizes for speed but still with a border mask). The best-fit $\hat{\phi}_J$ for each of these displays a qualitatively similar mean field, while their best-fit χ^2 appear to be in line with expectation as shown in Fig. 8. Finally, we check that even initializing Algorithm 1 at the true ϕ results in the same mean-field feature in $\hat{\phi}_J$ and a similar best-fit χ^2 value.

As the final piece of evidence that the mean field is a necessary feature of the joint MAP estimate of ϕ, we show that similar biases occur naturally in other MAP estimates for models which have more parameters than data and thus yield highly non-Gaussian posteriors. Consider the

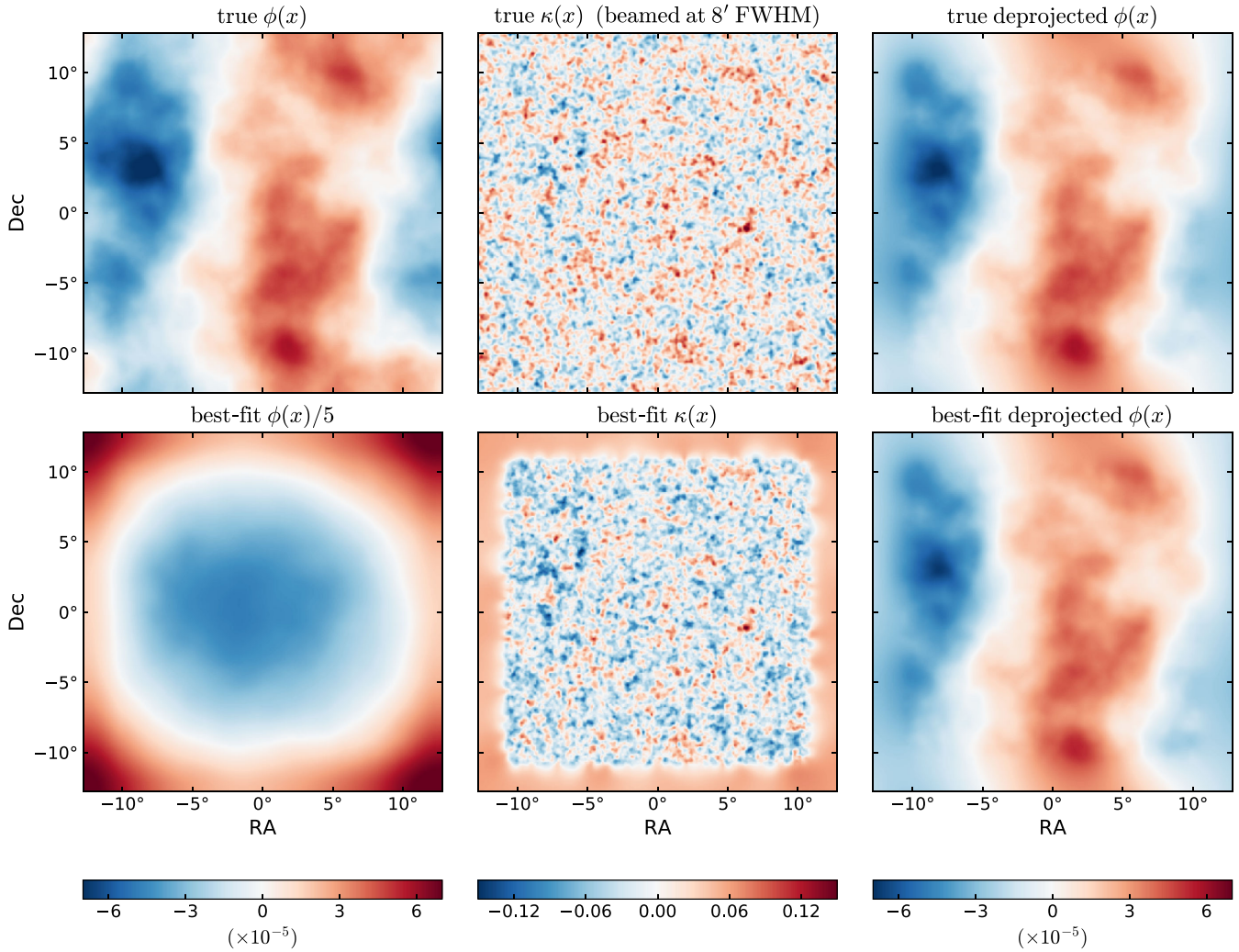

FIG. 6. The reconstructed lensing potential from a run of our algorithm on simulated data (bottom row), as compared to the simulation truth (top row). The first column is the raw $\phi(x)$ map that maximizes the posterior. The middle column is the corresponding convergence, $\kappa(x) \equiv -\nabla^2\phi(x)/2$, which allows one to see the good agreement with the truth in the unmasked regions. A small uniform negative "mean-field" correction inside the mask is visually recognizable as a slight preponderance of blue. The final column is after deprojecting this mean field using the procedure described in Sec. V B, allowing one to better recognize the agreement with the true ϕ map.

following toy example which is relevant to the problem of estimating scalar-to-tensor ratio r and which will foreshadow the discussion in the next section where we free r as a parameter.

Suppose we observe a noisy signal which is the product of some scaling parameter, r, with some Gaussian random field, B,

$$d = rB + n, \tag{48}$$

where n is stationary noise and n and B have known spectral densities $\mathcal{C}_n$ and $\mathcal{C}_B$, respectively. Notice that for a given value of r, the maximum of $B \mapsto \mathcal{P}(r, B|d)$ is given by a Wiener filterlike solution,

$$\hat{B}(r) \equiv r\mathcal{C}_B(\mathcal{C}_n + r^2\mathcal{C}_B)^{-1}d. \tag{49}$$

Therefore, the joint MAP estimate of r and B can be computed by maximizing $r \mapsto \mathcal{P}(r, \hat{B}(r)|d)$. However, a simple calculation shows that this function is always maximized at $r = 0$.[8] The cause of this singularity is simply that there is a perfect degeneracy in the likelihood term wherein one can decrease B and increase r and fit the data identically. The best fit of the full posterior will then maximize just the prior along this slice of parameter space,

[8]This statement depends on the prior one takes on r, e.g., the singularity is at $r = 0$ with a Jeffrey's prior as we have assumed here, but at $r = \infty$ with a flat prior. Nevertheless, no reasonable data-independent prior can remove the singularity entirely, which is the important part of our example.

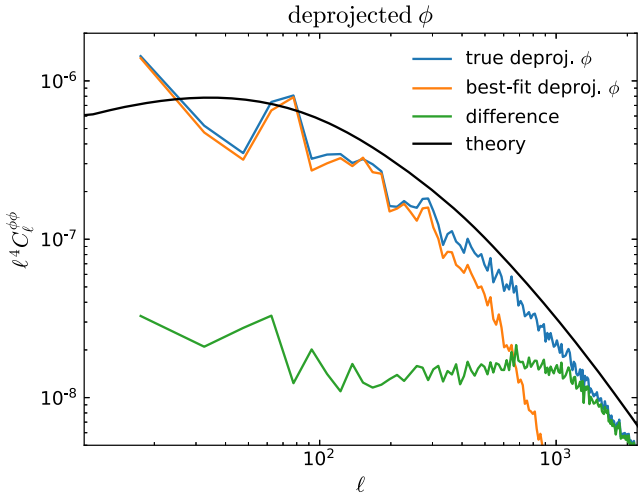

FIG. 7. The power spectra of the best-fit ϕ map as compared to the simulation truth and theory spectrum for the run with real-space masking described in Sec. V B. The best fit and simulation truth ϕ maps are the ones shown in the right column of Fig. 6 and have had the mean-field deprojected according to the procedure described in Sec. V B.

which in this case happens at $r = 0$. Yet, the posterior expected value of r, which effectively marginalizes over the unknown B, gives a perfectly normal and nonzero estimate of r. To complete the analogy, note the similarity in data residual between the lensing case and our toy example, $d - \mathbb{L}(\phi)f$ and $d - rB$. Thus, for similar reasons as in this toy example, the MAP estimate of $\hat{\phi}_J$ is driven away from its expected value, although due to the nonperfect degeneracy we are not driven all the way to any singularities at zero.

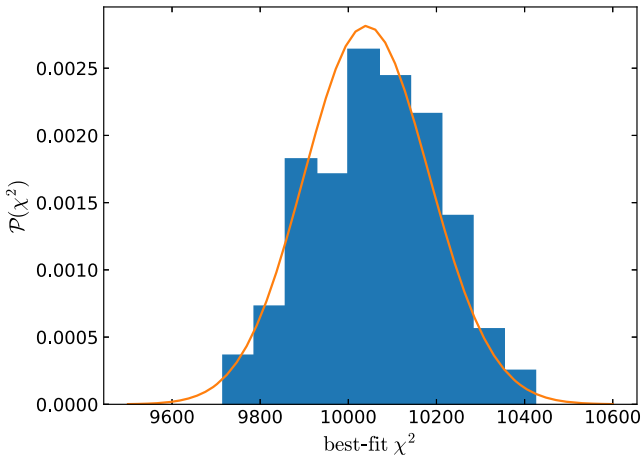

FIG. 8. Distribution of the χ^2 of the best-fit point from runs on 500 different simulated datasets. For speed, we have reduced the map size as compared to the main runs described in this work to 128×128 pixels (while keeping the relative width of the border mask width) and use only E and B. The expected distribution of the best-fit χ^2 under a Gaussian approximation of the posterior is shown as the orange curve.

Our point with this example is to demonstrate that MAP estimates need not be optimal, and to stress that while MAP estimates can have poor properties as estimators (such as in this case for r), sampling the posterior will always yield the correct answer. Nevertheless, the fact that $\hat{\kappa}_J$ tracks fluctuations of κ with little apparent bias suggests $\hat{\kappa}_J$ could still form a useful estimator, and moreover potentially be more useful for initializing a sampling algorithm for the joint posterior.

C. With r as a free parameter

The toy example from the previous section serves a dual purpose, as it was selected to prepare discussion of the actual problem of r estimation. The differences are that in reality we have tensor contributions to T and E in addition to just B, and of course because the toy example did not involve lensing. Nevertheless, we might expect qualitatively similar behavior, and in this section we verify that this is indeed the case.

To do so, we generate simulated data with $r = 0.05$ then run the maximization algorithm for $\mathcal{P}(f, \phi|d, r)$ over a grid of r values from $r = 0$ to $r = 0.15$. More specifically, we compute,

$$(\hat{f}(r), \hat{\phi}(r)) = \mathrm{argmax}_{f,\phi} \mathcal{P}(f, \phi|d, r), \tag{50}$$

and plot the function $r \mapsto \mathcal{P}(r, \hat{f}(r), \hat{\phi}(r)|d)$ as the blue curve in Fig. 9. Indeed we find that a singularity at zero

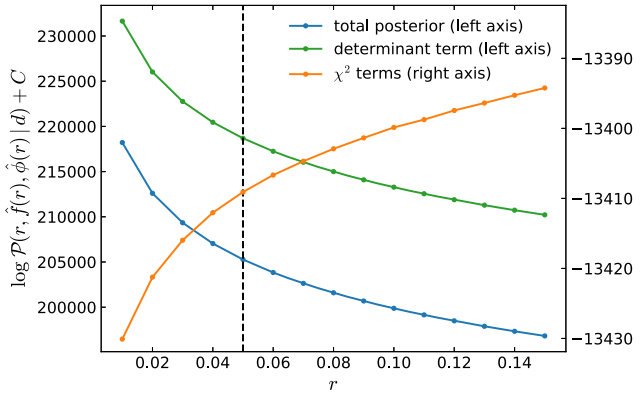

FIG. 9. A slice through the joint posterior probability (3), varying r and maximizing with respect to f and ϕ for each value of r. For speed, we have reduced the map size as compared to the main runs described in this work to 128×128 pixels (while keeping the relative width of the border mask the same). The green curve (left axis) is the contribution from $\det \mathbb{C}_f(\theta)$, the orange curve (right axis) is the contribution from the three χ^2 terms [i.e., the first three terms of (3)], and the blue (left axis) is the sum of these two. This demonstrates that the joint MAP estimate of r is not useful as it is driven to zero. The lack of apparent numerical noise in the orange curve demonstrates the stability of the maximization algorithm.

exists, which confirms that the MAP estimate of r (jointly with f and ϕ) is not a useful estimator, as it is always zero.

We point out that the total posterior plotted in blue is largely dominated by just the determinant of the CMB covariance in (3), $\det \mathbb{C}_f(r)$. This is independent of f and ϕ and hence independent of the maximization algorithm; to see the performance of the maximization, we plot in orange the contribution to the total posterior from only χ^2 terms, i.e., the first three terms of (3). The smoothness of this curve is further evidence of the quality of convergence, as we might otherwise expect to see lots of numerical noise in adjacent bins.

This convergence is important because the orange curve gives one contribution to the full marginal posterior, $\mathcal{P}(r)$, and if this piece were not stable numerically, adding in the other contributions would be of no use. Indeed, under the Laplace approximation we can compute the marginal posterior by just adding in a determinant term, i.e., the analog of the denominator in (11) but for marginalization over *both* f and ϕ, and which would cancel out the singularity seen here. In fact, something like this could potentially be calculable in practice with Hessian operators and if one can compute accurately enough the necessary determinant via Monte Carlo. Ultimately, we seek to sample directly from the exact posterior, producing a marginal $\mathcal{P}(r)$ with no approximation. Again, the stability of the curves in Fig. 9 suggest this should be numerically possible as long as satisfactory convergence of the sampling algorithm can be achieved.

VI. VALIDATION OF LENSED POSTERIOR APPROXIMATION

Having worked throughout this paper using the posterior approximation in Eq. (5), we now explain why this is an adequate approximation to the true posterior given in Eq. (4). This explanation will make use of the machinery built up thus far to compute $\hat{\phi}_J$, hence why we have delayed it up until this point.

The exact unlensed posterior is given in Eq. (3); if we could work with this, there would be no need to use and validate Eq. (5). Indeed, it may be tempting to simply perform a change of variables in Eq. (3) to $\tilde{f} = \mathbb{L}(\phi) f$, which introduces a Jacobian determinant term given by $\det \mathbb{L}(\phi)$, and then use the proof that the LenseFlow determinant is unity to ignore this term. However, it is important to note that $\mathbb{L}(\phi)$ here is an infinite-dimensional operator, and in general infinite-dimensional determinants are ill defined; indeed, the proof of unit determinant for LenseFlow does not necessarily extend trivially down to infinitely small pixels.[9] As such, in our discussion, we must only appeal to determinants of finite-dimensional operators, which are instead completely well defined.

We can do so by considering the posterior for the unlensed fields on an extremely high but finite resolution, q (we will denote these fields as f_q), which will be far higher than the data resolution, p. That is, by considering the data model

$$d = \mathbb{P}\mathbb{L}_q(\phi) f_q + n, \tag{51}$$

where $\mathbb{L}_q$ is a lensing operation on the q pixelization and $\mathbb{P}$ further pixelizes the q resolution to p. Unlike in Eq. (3), with the posterior on (f_q, ϕ), we are free to perform the change of variables $\tilde{f}_q = \mathbb{L}_q(\phi) f_q$ and ignore the resulting determinant as long as we use LenseFlow for $\mathbb{L}_q$, as here we are considering a finite pixelization.

Furthermore, if we make the q resolution fine enough, the resulting reparametrized posterior must converge to

$$\begin{aligned} &-2\log \mathcal{P}(\tilde{f}_q, \phi, \theta | d) \\ &= (d - \mathbb{P}\tilde{f}_q)^\dagger \mathbb{C}_n^{-1} (d - \mathbb{P}\tilde{f}_q) \\ &\quad + \tilde{f}_q^\dagger \mathbb{L}_q^{-\dagger}(\phi) \mathbb{C}_{f_q}(\theta) \mathbb{L}_q^{-1}(\phi) \tilde{f}_q + \log\det \mathbb{C}_{f_q}(\theta) \\ &\quad + \phi^\dagger \mathbb{C}_\phi(\theta)^{-1} \phi + \log\det \mathbb{C}_\phi(\theta). \end{aligned} \tag{52}$$

Note that if instead we make q coarser until it is the same as p, this becomes exactly the approximate posterior, Eq. (3), which we are trying to validate.

The path to validating Eq. (3) is thus the following. Compute $\hat{\phi}_J$ from Eq. (52) where q is a very fine resolution, and the simulated data configuration is similar to the other cases discussed in this paper. This will asymptote to the exact answer as we reduce the q pixelization, and we verify it indeed asymptotes as we vary q from $1/2$ to $1/8$ of the data pixelization. With this in hand, we now compare against the $\hat{\phi}_J$ we get from Eq. (3). In all cases explored, we find that the power spectrum of the difference between the two estimates of ϕ is roughly three orders of magnitude below the spectrum of $\hat{\phi}_J$ itself at all angular scales below the data Nyquist frequency. We thus conclude Eq. (3) is a very good approximation.

We further note that, while we have checked that Eq. (3) is a good approximation when LenseFlow is used, other pixelized lensing approximations could conceivably be used as well (e.g., any of [31–34]). One would simply need to perform the same procedure laid out here, but using those algorithms instead to perform lensing on the p resolution. However, one would always have to use LenseFlow to perform the reconstruction on the high resolution q pixels, lest a determinant be introduced there (in the following section, we show that the determinant of at least one other pixelized lensing algorithm is different enough from unity that it matters). Thus, the development of LenseFlow is the crucial piece of this proof.

[9] An earlier version of this work claimed precisely this, which may or may not be true. We do not attempt to give a proof of determinant of lensing in the infinite resolution here.

VII. NUMERICAL LENSING DETERMINANTS

In Sec. IV we have proven LenseFlow has unit determinant on any finite pixelization in the absence of ODE integration errors. Here, we check the typical determinant values achievable with finite number of ODE steps, as well as checking the determinant of one other lensing algorithm.

For relatively small numbers of pixels, it is computationally feasible to compute the determinant for a given algorithm by explicitly calculating the matrix representation of $\mathbb{L}(\phi)$ for a given ϕ and taking its determinant.[10] We have done so for map sizes between 8×8 and 64×64, and for the standard approximation to lensing where one expands in a Taylor series around the deflection,

$$\tilde{f}(x) = f(x + \nabla\phi(x)) = f(x) + \nabla^i\phi(x)\nabla^i f(x) + \cdots \quad (53)$$

To minimize the error incurred by the Taylor series approximation due to pixelization, we have performed the test here with 1 arcmin pixels, i.e., somewhat smaller than the 3 arcmin pixels we use in the rest of this paper. For this pixel size, the determinant of the Taylor series lensing approximation asymptotes by the 7th order term in the expansion. By using this many terms, we are testing the determinant due to the implicitly assumed subpixel extrapolation method of the Taylor series expansion, rather than the determinant due to Taylor series truncation error.

The exact value of the determinant is, in fact, an unimportant overall normalization factor; instead, what is important is how it varies as a function of ϕ near the peak of the probability distribution as compared to the other terms in the posterior probability. As a simple way to mimic samples of ϕ near this peak, we approximate the problem as a Wiener filter problem, and use the analytic calculation of the effective reconstruction noise, $\mathbb{N}_\phi$, from the iterated full sky quadratic estimator [10]. We expect the determinant will be most important when the effective noise is high, such as when performing a temperature-only reconstruction; since we want our method to work for these cases, we check using the temperature-only $\mathbb{N}_\phi$. Finally, we have not upscaled the reconstruction noise for our smaller $f_{\rm sky}$, thus this check will represent a lower bound on how important the determinant might be. To mimic the samples of ϕ, we first simulate a one single typical best-fit (i.e., "Wiener filtered") ϕ, which is given from the covariance $\mathbb{C}_\phi(\mathbb{C}_\phi + \mathbb{N}_\phi)^{-1}\mathbb{C}_\phi$. We then simulate many samples from around the peak which are given by an additive contribution drawn from $\mathbb{C}_\phi(\mathbb{C}_\phi + \mathbb{N}_\phi)^{-1}\mathbb{N}_\phi$. For each of these samples, we calculate the prior and lensing determinant terms in (5). We consider the scatter in the prior term a proxy for the level of change we might be able to tolerate, and this should be a fairly good proxy since this term dominates the posterior at the smallest scales to which we expect the determinant to be most sensitive. Figure 11 shows the results. We find that the determinant term varies roughly on the same order as the prior term, even sometimes larger. Hence it does not appear that it can be ignored for Taylens, at least not on the scales probed by these maps (which are, indeed, relevant physical scales in general). It is thus necessary that we use LenseFlow

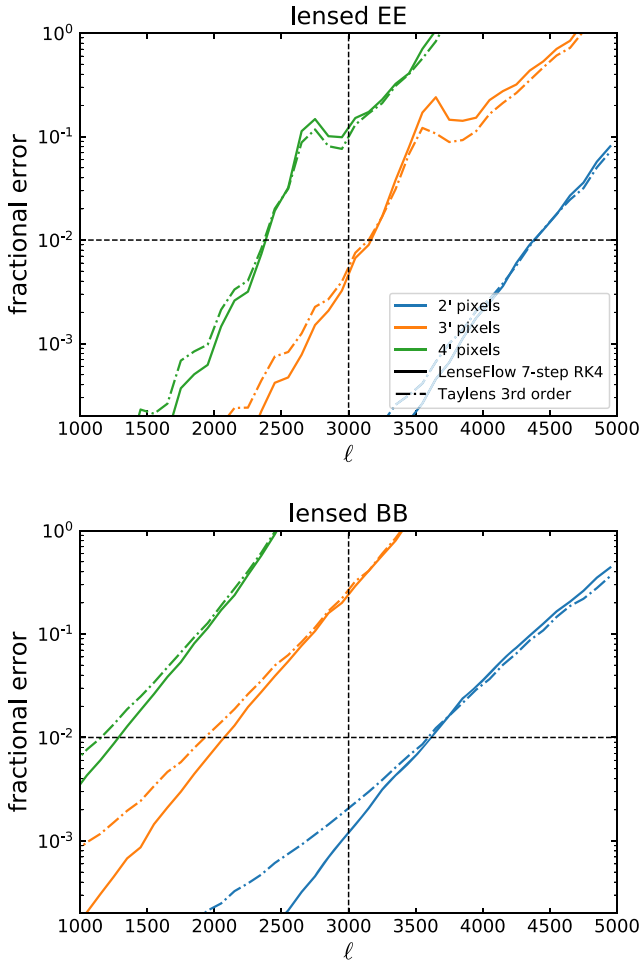

FIG. 10. Comparison of the accuracy of LenseFlow vs Taylor series lensing augmented with a nearest-pixel permutation (i.e., "Taylens" [35]) The quantity plotted is the spectrum of $\mathbb{P}\mathbb{L}_{\rm true}(\phi)f - \mathbb{L}(\mathbb{P}\phi)\mathbb{P}f$ divided by the theoretical EE or BB spectrum, where P is a pixelization operation and $\mathbb{L}_{\rm true}$ is an asymptotically high order LenseFlow. The first term represents the "true" lensed field, while the latter performs lensing on a pixelization map. For the "true" fields, we use pixels 4 times smaller on a side than the pixelized version. This figure demonstrates that our use of 7-step LenseFlow on 3 arcmin pixels incurs an error of ~20% in BB by $\ell = 3000$, which is far below the instrumental noise at this multipole for, e.g., CMB-S4. The error is approximately the same as using Taylens, showing that the error is driven by the loss of information due to pixelization rather than due to an error in ODE integration or due to Taylor series truncation.

[10]This can be done by applying the operator to some set of maps which form a complete basis. It may also be possible to use other methods to compute the determinant; we have chosen this route only for simplicity.

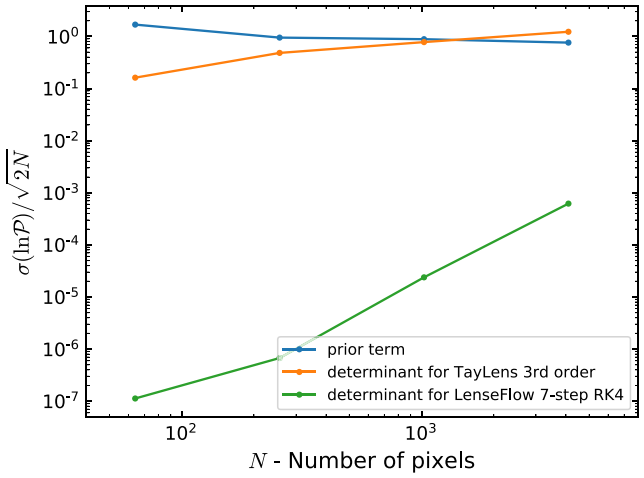

FIG. 11. The standard deviation of the variation in the log probability values for the ϕ-prior term, $\phi^\dagger \mathbb{C}_\phi^{-1}\phi$, and lensing determinant term, $2\log|\det\mathbb{L}(\phi)|$, in (5), as computed from Monte Carlo samples of ϕ. These samples approximate samples from the posterior probability $\mathcal{P}(f,\phi|d)$ for some simulated data, d, assuming full-sky temperature-only reconstruction noise. Here we have used 7th order Taylor series lensing on 1 arcmin pixels with temperature-only data. Because the variation in the two terms is of similar order, the determinant cannot be ignored.

as our lensing operator, as its determinant can be made arbitrarily close to one. We find that solving the LenseFlow ODE with seven steps of 4th order Runge-Kutta integration yields a determinant many orders of magnitude below the variation in the ϕ-prior term, and additionally yields acceptable errors on the lensed spectra as demonstrated in Fig. 10.

VIII. CONCLUSIONS AND FUTURE WORK

In this work, we have presented the first algorithm which produces the joint MAP estimate of ϕ, f and cosmological parameters like r. There are two important aspects to the algorithm. First, a change of variables from the unlensed field, f, to the lensed one, $\tilde{f}$, greatly reduces the correlations in the posterior making maximization work much more efficiently. Second, the maximization is a coordinate descent over $\tilde{f}$ and ϕ, which breaks the problem into two clean pieces, one a robustly solvable Wiener filter problem and the other entirely independent of the instrument and data.

The workability of the algorithm depends on using a new lensing algorithm which we have developed called LenseFlow, which has determinant equal to unity, and allows us to trivially perform the aforementioned change of variables. LenseFlow appears unique among known algorithms in having such a simple determinant; although the only other determinant we have explicitly verified is for the Taylor series approximation, it seems unlikely that other algorithms would have this property without it having been constructed intentionally. Nevertheless, it is worth checking other algorithms as perhaps their determinant is close enough to unity that it can be ignored, in which case there could be benefits of speed or convenience to using them instead.

Independently of how we have used it here, LenseFlow is interesting theoretically as a new formulation of lensing. To date, it has clearly been a very useful tool for cosmologists to work with the Taylor series expansion for weak lensing; we would argue that the ODE expansion presented here should be a valuable addition to any cosmologists' "toolbox" as well, as it can in some cases be quite advantageous to work with. For example, we have used it to give a simple proof of the area-preserving nature of the pixelized lensing algorithm. Additionally, it is very convenient that inverses and adjoints are so easily calculated with LenseFlow, not just for lensing but also for the Jacobian and Hessian operators. Some of these are possible to calculate with other identities (e.g., [12]), but the LenseFlow solution is very straightforward conceptually, and is highly numerically stable.

We have also discussed the relationship between the joint posterior, $\mathcal{P}(f,\phi|d,r)$, and the marginal posterior, $\mathcal{P}(\phi|d,r)$, the latter which is the basis of the algorithm given by [12]. The two estimates $\hat{\phi}_J$ and $\hat{\phi}_M$ differ from each other by a mean-field correction, as do the corresponding delensed estimates $\hat{f}_J$ and $\hat{f}_M$, and we have elucidated the relation between all of these quantities in the context of a Laplacian integration. One is free to take any of these quantities as an estimator and debias and quantify its uncertainties via simulations, and this would certainly lead to improvement over the quadratic estimate. However, any such procedure would suffer from the problem of needing to assume a value for r for these simulations, and perhaps from requiring too large a computational cost, so it is unclear if that is the right way to proceed forward.

Another approach is to compute the mean ϕ and f and their uncertainties by obtaining samples from the posterior via Markov-Chain Monte Carlo techniques. To be efficient, any such sampling algorithm likely needs to evaluate the gradient of the posterior at each sampled point. When sampling the marginalized posterior $\mathcal{P}(\phi,r|d)$, this gradient has a contribution from a determinant term which must be computed by averaging over simulations, and this likely becomes computationally too costly on an inner loop of a sampling algorithm. Additionally, the determinant makes it impossible to evaluate the value of posterior at a given point, which is necessary for, e.g., accept-reject steps. This may ultimately make it impossible to use the marginalized posterior for sampling, or force the requirement of very sophisticated sampling algorithms which do not need this quantity. Conversely, the joint posterior $\mathcal{P}(f,\phi,r|d)$ does not have such a determinant, allowing easy evaluation of the posterior value and of the gradient at any point. We thus consider the joint posterior the most optimistic path to pursue for sampling.

Insofar as the simulation assumptions presented in Sec. V are well approximated by the actual experimental conditions, the techniques presented in this paper can be immediately applied to real data. It is important to note, however, that proper application to real data should include a number of tests for robustness to things like uncertainty in cosmological parameters which are assumed fixed and known in our simulations, nonstationary noise and beam, post-Born and non-Gaussian contributions to the lensing potential, effects of curved sky, foregrounds, etc. With these issues under control, the resulting joint MAP estimate, along with its uncertainties quantified by simulations, provides a realistic way to perform an analysis more optimal than one based on the quadratic estimate. Of course, other approaches exist as well, including other estimators and posterior sampling; ongoing work will ascertain which method exactly is most useful in practice.

ACKNOWLEDGMENTS

We thank Antony Lewis and Thibaut Louis for helpful discussions during the course of this work, and Julien Carron for sharing his insights on the lensing determinant. E. A. is supported in part by NSF CAREER DMS-1252795, DMS-1812199, an IHES CARMIN Fellowship, and a University of California Davis Chancellor's Fellowship. M. M. and B. D. W. are supported by the Labex ILP (reference ANR-10-LABX-63). The work of B. D. W. is supported by the Simons Foundation.

[1] B. D. Sherwin, J. Dunkley, S. Das *et al.*, Phys. Rev. Lett. **107**, 021302 (2011).
[2] K. N. Abazajian, K. Arnold, J. Austermann *et al.*, Astropart. Phys. **63**, 66 (2015).
[3] K. N. Abazajian, P. Adshead, Z. Ahmed *et al.*, arXiv:1610.02743.
[4] D. Green, J. Meyers, and A. van Engelen, J. Cosmol. Astropart. Phys. 12 (2017) 005.
[5] W. Hu and T. Okamoto, Astrophys. J. **574**, 566 (2002).
[6] T. Okamoto and W. Hu, Phys. Rev. D **67**, 083002 (2003).
[7] D. Hanson, A. Challinor, and A. Lewis, Gen. Relativ. Gravit. **42**, 2197 (2010).
[8] M. Kesden, A. Cooray, and M. Kamionkowski, Phys. Rev. Lett. **89**, 011304 (2002).
[9] L. Knox and Y.-S. Song, Phys. Rev. Lett. **89**, 011303 (2002).
[10] K. M. Smith, D. Hanson, M. LoVerde, C. M. Hirata, and O. Zahn, J. Cosmol. Astropart. Phys. 06 (2012) 014.
[11] C. M. Hirata and U. Seljak, Phys. Rev. D **68**, 083002 (2003).
[12] J. Carron and A. Lewis, Phys. Rev. D **96**, 063510 (2017).
[13] E. Anderes, B. D. Wandelt, and G. Lavaux, Astrophys. J. **808**, 152 (2015).
[14] J. Bezanson, A. Edelman, S. Karpinski, and V. Shah, SIAM Rev. **59**, 65 (2017).
[15] D. Beck, G. Fabbian, and J. Errard, Phys. Rev. D **98**, 043512 (2018).
[16] V. Böhm, B. D. Sherwin, J. Liu, J. Colin Hill, M. Schmittfull, and T. Namikawa, Phys. Rev. D **98**, 123510 (2018).
[17] A. L. Caterini and D. E. Chang, arXiv:1608.04374.
[18] M. H. Abitbol, Z. Ahmed, D. Barron *et al.*, arXiv:1706.02464.
[19] D. Barron, Y. Chinone, A. Kusaka *et al.*, J. Cosmol. Astropart. Phys. 02 (2018) 009.
[20] B. D. Sherwin, A. van Engelen, N. Sehgal *et al.*, Phys. Rev. D **95**, 123529 (2017).
[21] K. T. Story, D. Hanson, P. A. R. Ade *et al.*, Astrophys. J. **810**, 50 (2015).
[22] P. A. R. Ade, N. Aghanim *et al.* (Planck Collaboration), Astron. Astrophys. **594**, A13 (2016).
[23] N. Aghanim, M. Ashdown *et al.* (Planck Collaboration), Astron. Astrophys. **596**, A107 (2016).
[24] P. A. R. Ade *et al.* (BICEP2 and Keck Array Collaborations), Phys. Rev. Lett. **116**, 031302 (2016).
[25] F. Elsner and B. D. Wandelt, Astron. Astrophys. **549**, A111 (2013).
[26] K. M. Huffenberger, Mon. Not. R. Astron. Soc. **476**, 3425 (2018).
[27] D. Kodi Ramanah, G. Lavaux, and B. D. Wandelt, Mon. Not. R. Astron. Soc. **468**, 1782 (2017).
[28] D. S. Seljebotn, K.-A. Mardal, J. B. Jewell, H. K. Eriksen, and P. Bull, Astrophys. J. Suppl. Ser. **210**, 24 (2014).
[29] K. M. Smith, O. Zahn, and O. Dore, Phys. Rev. D **76**, 043510 (2007).
[30] E. Anderes, Ann. Stat. **38**, 870 (2010).
[31] S. Hamimeche and A. Lewis, Phys. Rev. D **77**, 103013 (2008).
[32] G. Lavaux and B. D. Wandelt, Astrophys. J. Suppl. Ser. **191**, 32 (2010).
[33] A. Lewis, Phys. Rev. D **71**, 083008 (2005).
[34] T. Louis, S. Næss, S. Das, J. Dunkley, and B. Sherwin, Mon. Not. R. Astron. Soc. **435**, 2040 (2013).
[35] S. K. Næss and T. Louis, J. Cosmol. Astropart. Phys. 09 (2013) 001.