Supplementary Materials

Yunqing Ouyang (欧阳云卿),[1,2] Qing-Rui Wang (王晴睿),[3,*] Zheng-Cheng Gu (顾正澄),[4,†] and Yang Qi (戚扬)[1,2,5,‡]

[1] *State Key Laboratory of Surface Physics, Fudan University, Shanghai 200433, China*
[2] *Center for Field Theory and Particle Physics, Department of Physics, Fudan University, Shanghai 200433, China*
[3] *Department of Physics, Yale University, New Haven, CT 06511, USA*
[4] *Department of Physics, The Chinese University of Hong Kong, Shatin, New Territories, Hong Kong, China*
[5] *Collaborative Innovation Center of Advanced Microstructures, Nanjing 210093, China*

(Dated: November 17, 2021)

CONTENTS

I. MOTIVATION: $\mathbb{Z}_n$-SPT IN 2D

We first motivate our method using the task of identifying the cohomology class of a cocycle, with the example of a simple cyclic group $\mathbb{Z}_n$. In this case, cocycles in the simplified basis are related to the topological invariants used in previous studiess [1].

In particular, we consider the task of checking whether the result of an obstruction function is a trivial or nontrivial cocycle. (For the definition of cocycles and their cohomology classes, see Sec. II A.) Instead of using coboundary equations for the inhomogeneous cochains directly, which results in a large computational cost, we construct topological invariants. In general, a $(d+1)$-cocycle $\alpha \in H^{d+1}[G, \mathrm{U}(1)]$ can be interpretted as a d-dimensional bSPT state. In fact, such a cocycle can be used to construct partition functions on any closed $(d+1)$D manifold M, with arbitrary symmetry fluxes of G inserted in noncontractible loops of M [2]. Such a combination of closed (3+1)D manifold and symmetry fluxes is knwon as a G-bundle. A trivial 4-cocycle, representing a trivial SPT phase, gives a partition function that evaluates to the trivial value of $+1$ on any G-bundle; a nontrivial cocycle, on the other hand, evaluates to nontrivial values on some nontrivial G-bundles.

Furthermore, only a few number of representative G-bundles need to be checked, each detecting one root cohomology class in $H^{d+1}[G, \mathrm{U}(1)]$. If the partition function is trivial on all these G-bundles, the corresponding cocycle is trivial.

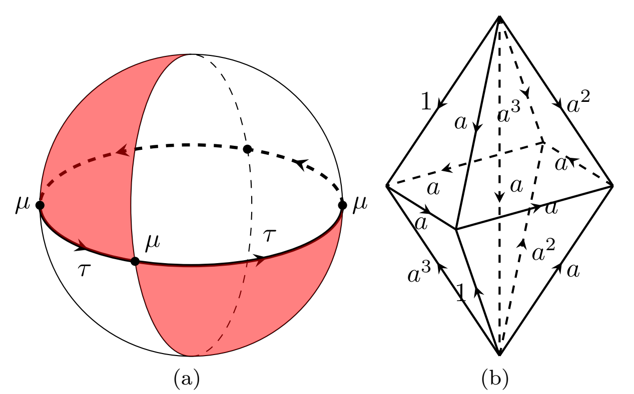

FIG. 1. The lens space $L_4(1)$. (a) $L_4(1)$ constrcuted by gluing the upper and lower hemispheres of a 3-ball. The upper hemisphere is first rotated by 90 degrees before glued to the lower hemisphere, as indicated by the red marks on both hemispheres, which are glued together. $L_4(1)$ has one 3-cell, which is the interior of the ball; it has one 2-cell, which is the upper hemisphere or the lower hemisphere (they are identified during the gluing); it has one 1-cell, which is one of the segments τ on the equator shown in thick lines with arrows (the equator is divided into four segments, which are identified with each other after the gluing); it has one 0-cell, which is the starting and the ending point of τ and labeled by μ. (b) A triangulation (Δ-complex decomposition) of $L(4,1)$, and a flat connection realzing the nontrivial symmetry flux along τ.

To demonstrate this procedure, we consider (2+1)D

* Present address: Yau Mathematical Sciences Center, Tsinghua University, Beijing 100084, China
† zcgu@phy.cuhk.edu.hk
‡ qiyang@fudan.edu.cn

manifolds, which are easy to illustrate, and a simple symmetry group: the cyclic group $G = \mathbb{Z}_n = \langle a|a^n = 1\rangle$. This example also appeared in Ref. [1]. For this simple case, the (2+1)D bSPT phases are classified by $H^3[G, \mathrm{U}(1)] = Z_n$. Hence, there is only one root state, and one corresponding representative G-bundle. This G-bundle is illustrated in Fig. 1 for the case of $n = 4$. The base manifold of this G-bundle is constructed by starting from a solid 3-ball and gluing the two hemispheres on the surface of the ball in the following twisted way: the upper hemisphere is rotated by an angle of $2\pi/n$, reflected with respect to the equator, and glued to the lower hemisphere. Consistent with this gluing, the equator can be divided evenly into n segments, which are identified with each other. Consequently, the starting and end points of each segment are also identified as the same point, and the segment becomes a noncontractible loop. The gluing creates a closed 3-manifold M, which is known as the lens space $L_n(1)$ in mathematics [3]. This manifold has a nontrivial first homotopy group $\pi_1(M) = \mathbb{Z}_n$, generated by the noncontractible loop τ shown on Fig. 1(a). The G-bundle has a nontrivial symmetry flux a (a labels the generator of the $\mathbb{Z}_n$ group) along this loop.

We now evaluate on this G-bundle the partition function constructed from a 3-cocycle $\alpha \in H^3[G, \mathrm{U}(1)]$. We assume that α is computed as an inhomogeneous cocycle. As explained in details in Sec. V, an inhomogeneous 3-cocycle can be used to construct partition functions on a simplicial complex with a flat gauge connection, which is basically a triangulated space consists of many tetrahedra (3-simplices). The gauge connection consists of $g_{ij} \in G$ assigned to each edge $[v_i v_j]$ in the complex, satisfying two constraints: First, the total flux going around a triangle $[v_i v_j v_k]$ must vanish: $g_{ij} g_{jk} = g_{ik}$. Second, the total flux going around a noncontractible loop in $\pi_1(M)$ must produce the assigned symmetry flux in the G-bundle. On such a simplicial-complex realization of the G-bundle, a partition function of the SPT phase represented by the cohomology class α is constructed by multiplying weights associated with each tetrahedron: on one tetrahedron, denoted by its four vertices as $[v_0 v_1 v_2 v_3]$, the weight is given by

$$\exp\{\pm 2\pi i \langle \alpha, [v_0 v_1 v_2 v_3] \rangle\} = \exp\{\pm 2\pi i \alpha(g_{01}, g_{12}, g_{23})\}. \tag{1}$$

Here, the overall sign in the phase is plus (minus) if the orientation of the simplex is positive (negative), respectively.

Therefore, in order to evaluate the partition function on the G-bundle in Fig. 1(a), we must first decompose it into a simplicial complex, and assign a choice of flat connection g_{ij}. One particular construction is given in Fig. 1(b). It is then straightforward to compute the partition function:

$$Z = \exp\left\{ 2\pi i \sum_{j=1}^{n} \alpha(a, a^j, a) \right\}. \tag{2}$$

This partition function detects the classification of the SPT states: the trivial SPT phase gives $Z = +1$, while the root state of nontrivial SPTs gives $Z = e^{i2\pi/n}$. Therefore, it can be used as a topological invariant to determine the cohomology class of the cocycle α. In general, the value of Z can be $Z = e^{i2\pi k/n}$, where $k = 0, 1, \ldots, n-1$ indicates the cohomology class of α. In practice, the cohomology class of an inhomogeneous cocycle can be determined by evaluating such topological invariants instead of solving the cocycle equations of the inhomogeneous cocycles, which is a time-consuming task. In the rest part of the paper, we will introduce automated procedures to construct such topological invariants for generic discrete groups.

II. ALGEBRAIC DESCRIPTION

In this section, we describe a general algorithm for constructing topological invariants. Physically, the topological invariants are constructed from evaluating the partition functions on representative G-bundles. Hence, the algorithm contains two parts: First, one chooses representative G-bundles by constructing the classifying space of G, denoted by BG. Second, a triangulation of the G-bundles is computed by constructing a cellular map from BG to a standard simplicial realization of BG. These two steps are discussed in Secs. II A and II B, respectively. Finally, in Sec. II C, we combine the two steps and construct an algorithm for constructing the invariants that check the trivialness of a cocycle. Although the algorithm has a nice interpretation in terms of evaluating SPT partition functions on G-bundles, the derivation of the chain map can be described purely algebraically. For conciseness, we only discuss the algebraic construction of the algorithm in this section, and defer the discussion of physical interpretation to Sec. IV.

A. Classifying space and resolution

The classifying space of G, denoted by BG, is a topological space satisfying the following conditions: its first homotopy group (or the fundamental group) is $\pi_1(BG) = G$, and all its higher homotopy groups vanish: $\pi_k(BG) = 0$, $k > 1$. Closely related to BG, the universal bundle EG, is also the universal cover of BG. Since the fundamental group of BG is G, EG can be viewed as a topological space with a free action of G, and BG is the quotient space $BG = EG/G$.

EG is called the universal bundle, because any G-bundle can be constructed as a pullback bundle from its base space BG. As a result, a cohomology class on BG can be used to define partition functions on all possible G-bundles. This leads to the conclusion that d-dimensional bSPT phases are classified by $H^{d+1}[BG, \mathrm{U}(1)] = H^{d+1}[G, \mathrm{U}(1)]$. In fact, in the real computation, we do not need all geometric details of BG and EG. Instead, only the cellular chain complex of EG

is needed. Here, we review the algebraic structure of this chain complex, which is also known as a free $\mathbb{Z}G$-resolution (of $\mathbb{Z}$).

Mathematically, we construct EG as a CW-complex, which is a model of topological spaces widely used in algebraic topology, especially in the theory of singular homology. The precise definition of a CW-complex can be found in Appendix A of Ref. [3]. Roughly speaking, a CW-complex is made by gluing cells of different dimensions, where each d-dimensional cell, or a d-cell for short, is homeomorphic to a d-dimensional disk. We denote the collection of d-cells in the CW-complex EG as $(EG)_d$.

In singular-homology theory, a d-chain is a formal summation of d-cells, with integral coefficients. Hence, the space of d-chains, denoted by $C_d(EG)$, is a $\mathbb{Z}$-module with basis in $(EG)_d$. Since G has a free action on EG, the modules $C_d(EG)$ are actually free $\mathbb{Z}G$-modules. Furthermore, they form the following long exact sequence under the boundary map,

$$\cdots \to C_k(EG) \xrightarrow{\partial_k} C_{k-1}(EG) \to \cdots \to C_1(EG) \xrightarrow{\partial_1} C_0(EG) \xrightarrow{\epsilon} \mathbb{Z} \to 0. \tag{3}$$

This long exact sequence is known as the augmented chain complex of EG.

In practice, we only need to keep track of the algebraic structure of the chain complex above. From this view point, we have free $\mathbb{Z}G$-modules $F_d = C_d(EG)$ forming a long exact sequence,

$$\cdots \to F_k \xrightarrow{\partial_k} F_{k-1} \to \cdots \to F_1 \xrightarrow{\partial_1} F_0 \xrightarrow{\epsilon} \mathbb{Z} \to 0. \tag{4}$$

This is called an augmented free $\mathbb{Z}G$-resolution.

The exactness of the sequences in Eqs. (3) and (4) follows the fact that the space EG is contractible. Mathematically, this means that the identity map from EG to itself is homotopic to the zero map that maps EG to an empty space. Such a homotopy equivalence between these two maps is called a contracting homotopy, and it plays an essential role in the construction of chain maps in Sec. II B. Algebraically, a contracting homotopy s is a collection of $\mathbb{Z}$-linear maps from each module F_k to the module in one higher dimension, F_{k+1}, as shown in the following diagram:

$$\begin{tikzcd}
F_{k+1} \arrow[r, "\partial_{k+1}"] & F_k \arrow[r, "\partial_k"] \arrow[d, "\mathrm{id}"] \arrow[dl, "s_k"] & F_{k-1} \arrow[dl, "s_{k-1}"] \\
F_{k+1} \arrow[r, "\partial_{k+1}"] & F_k \arrow[r, "\partial_k"] & F_{k-1}
\end{tikzcd} \tag{5}$$

This is not a commutative diagram. Instead, the maps satisfy the following condition,

$$\partial_{k+1}s_k + s_{k-1}\partial_k = \mathrm{id}\,. \tag{6}$$

In other words, the anticommutator between s and ∂ is id, which can be understood as the difference between the identity map and the zero map. Hence, this indicates that s is a homotopy between these two maps. We emphasize that s is $\mathbb{Z}$-linear but not $\mathbb{Z}G$-linear in general, meaning that it does not commute with group action: $s_k(g \cdot x) \neq g \cdot s_k(x)$.

Algebraically, Eq. (6) implies that s can be viewed as an "inverse" of the boundary map: For a closed k-chain $x \in F_k$, the condition $\partial_k x = 0$ simplifies Eq. (6) to $\partial_{k+1}s_k(x) = x$. Hence, $s_k(x)$ is a $(k+1)$-chain that borders x. This immediately proves the exactness of the sequence in (4), because every cycle x is a boundary of $s_k(x)$. This operation of finding the inverse of the boundary map using a contracting homotopy will also play a vital role in the construction of chain maps in Sec. II B.

Once a resolution is constructed for a group G, it can be used to compute the group-cohomology classification and the invariants of the cocycles. The k-cochains are defined as $\mathbb{Z}G$-linear maps from F_k to the coefficient module M, and space of k-cochains is denoted by $C^k(G,M) = \mathrm{Hom}_G(F_k, M)$. Here, the subscript G indicates that the cochains are invariant under the action of G:

$$\langle \alpha, gx \rangle = g\langle \alpha, x \rangle. \tag{7}$$

In this paper, we use greek letters to denote cochains. The bracket $\langle \alpha, x \rangle$ denotes evaluating the linear map α on the element $x \in F_k$. The result of the bracket is a coefficient $\langle \alpha, x \rangle \in M$, and $g\langle \alpha, x \rangle$ denotes the G-action on M.

The boundary map $\partial_k : F_k \to F_{k-1}$ naturally induces a coboundary map $d^{k-1} : C^{k-1}(G,M) \to C^k(G,M)$:

$$\langle d^{k-1}\alpha, x \rangle = \langle \alpha, \partial_k x \rangle. \tag{8}$$

Using the coboundary maps, we can define the k-cocycles, which are k-cochains satisfying $d^k\alpha = 0$, and the k-coboundaries, which are the coboundary of $(k-1)$-cochains, $\alpha = d^{k-1}\beta$. The spaces of k-cocycles and k-coboundaries are $Z^k(G,M) = \ker d^k$ and $B^k(G,M) = \mathrm{img}\, d^{k-1}$, respectively. The property that $\partial_k\partial_{k+1} = 0$, or the boundary of a boundary is empty, implies that $d^k d^{k-1} = 0$. This ensures that $B^k(G,M)$ is a submodule of $Z^k(G,M)$, and allows us to define the k-th cohomology of G as the quotient of the two modules,

$$H^k(G,M) = \frac{Z^k(G,M)}{B^k(G,M)} = \frac{\ker d^k}{\mathrm{img}\, d^{k-1}}. \tag{9}$$

We emphasize that the cochain space $C^k(G,M)$, the resulting spaces $Z^k(G,M)$ and $B^k(G,M)$ all depend explicitly on the choice of the resolution F. However, the resulting group-cohomology modules $H^k(G, M)$ do not depend on the choice of the resolution. More precisely speaking, group-cohomology modules computed using different resolutions are naturally isomorphic to each other.

In the rest of this section, we give two examples to demonstrate the concept of free resolutions and their contracting homotopy. In the first example, we show how the inhomogeneous cocycles, which are widely used in physics literatures, can be expressed using this language. In fact, in math literatures, the corresponding resolution is called the bar resolution [4], which we shall denote by $\bar{F}$. This type of resolution can be constructed for an arbitrary group G. In the resulution $\bar{F}$, the module $\bar{F}_k$ is spanned by the $\mathbb{Z}G$ basis of the following form, $[g_1|g_2|\cdots|g_k]$, where $g_i \in G$. The boundary operator is given as the following,

$$\partial_k[g_1|\cdots|g_k] = g_1[g_2|\cdots|g_k] + \sum_{i=1}^{k-1}(-1)^i[g_1|\cdots|g_{i-1}|g_i g_{i+1}|g_{i+2}|\cdots|g_k] + (-1)^k[g_1|\cdots|g_{k-1}]. \tag{10}$$

Using this basis, a k-cochain α is represented as a function $\langle\alpha, [g_1|\cdots|g_k]\rangle$. Rewritten as $\alpha(g_1, \ldots, g_k)$, this is the inhomogeneous cochain used in physics literatures. Eq. (10) gives the familiar coboundary operation of the inhomogeneous cochains,

$$\begin{aligned}(d^k\alpha)(g_1, \ldots, g_{k+1}) =& g_1\alpha(g_2, \ldots, g_{k+1}) + \sum_{i=1}^{k}(-1)^i\alpha(g_1, \ldots, g_i g_{i+1}, \ldots, g_{k+1}) \\ &+ (-1)^{k+1}\alpha(g_1, \ldots, g_k).\end{aligned} \tag{11}$$

The bar resolution has the following contracting homotopy $\bar{s}$:

$$\bar{s}_k(g_0[g_1|\cdots|g_k]) = [g_0|g_1|\cdots|g_k]. \tag{12}$$

We notice that, as expected, the map $\bar{s}_k$ does not commute with the G-action. It is straightforward to check that $\bar{s}$ satisfies the condition in Eq. (6). Hence, it is a contracting homotopy, which confirms that $\bar{F}_k$ forms a long-exact sequence. This contracting homotopy will be used in Sec. II B to map inhomogeneous cochains to other basis.

The bar resolution can be cumbersome to work with, since the number of $\mathbb{Z}G$ basis in each module $\bar{F}_k$ grows exponentially with k, $\mathrm{rank}_{\mathbb{Z}G}\,\bar{F}_k = |G|^k$. It is well known that one can slightly improve this by eliminating the basis elements where any one of the group element g_i is 1, the identity element of G. Equivalently, in terms of inhomogeneous cocycles, one can always use coboundary equivalence to set $\alpha(g_1, \ldots, g_k) = 0$ if any $g_i = 1$. The resulting resolution is called the normalized bar resolution in mathematical literatures. In the rest of this paper, we will use $\bar{F}$ and $\bar{s}$ to denote the normalized bar resolution of a group G and the associated contracting homotopy, respectively.

As an example, we examine the free resolution constructed by this algorithm for the $\mathbb{Z}_n$ group, which is the chain complex of the infinite-dimensional lens space [3]. In this resolution, each F_k is generated by only one $\mathbb{Z}G$-basis, denoted by e_k. The boundary operator is given as the following,

$$\begin{aligned}\partial e_{2k-1} &= (a-1)e_{2k-2}, \\ \partial e_{2k} &= \left(1 + a + a^2 + \cdots + a^{n-1}\right) e_{2k-1}.\end{aligned} \tag{13}$$

Here, a denotes the generator of $\mathbb{Z}_n$ satisfying $a^n = 1$. The algorithm in HAP also constructs the following contracting homotopy of this resolution.

$$\begin{aligned}s_{2k-1}\left(a^m e_{2k-1}\right) &= \delta_{m,n-1} e_{2k}, \\ s_{2k}\left(a^m e_{2k}\right) &= \left(1 + a + \cdots + a^{m-1}\right) e_{2k+1}.\end{aligned} \tag{14}$$

Again, the map s_k does not commute with the G-action.

B. Chain map

The resolution and its contracting homotopy constructed by HAP already allow us to do a wide ranges of group-cohomology calculations, including computing the classification of the group cohomology, and computing the cup and higher-cup products [5–7]. However, there are still functions of cocycles that can only be conveniently expressed using the inhomogeneous cochains [8, 9]. The reduced resolution can still help us simplify the computation of these functions: We first compute the cocycle functions using inhomogeneous cochains, then map the resulting inhomogeneous cocycles to the reduced resolution using a chain map, which we shall construct in this section. In general, the chain maps between the two resolutions allow us to map cocycles between the two basis. In the next section, we shall see that these chain maps can help us reduce the computational cost of calculating fSPT classifications.

A chain map f between two resolutions F and F', $f: F \to F'$, is a collection of $\mathbb{Z}G$-linear maps $f_k : F_k \to F'_k$, such that the following diagram commutes,

$$\begin{tikzcd}
\cdots \arrow[r, "\partial_{k+1}"] & F_k \arrow[r, "\partial_k"] \arrow[d, "f_k"] & F_{k-1} \arrow[r, "\partial_{k-1}"] \arrow[d, "f_{k-1}"] & \cdots \arrow[r, "\partial_1"] & F_0 \arrow[r, "\epsilon"] \arrow[d, "f_0"] & \mathbb{Z} \arrow[d, "\mathrm{id}"] \\
\cdots \arrow[r, "\partial'_{k+1}"] & F'_k \arrow[r, "\partial'_k"] & F'_{k-1} \arrow[r, "\partial_{k-1}"] & \cdots \arrow[r, "\partial'_1"] & F'_0 \arrow[r, "\epsilon'"] & \mathbb{Z}
\end{tikzcd} \tag{15}$$

Here, we describe an algorithm of constructing a chain map $f : F \to F'$ between two free $\mathbb{Z}G$-resolutions, using a contracting homotopy s' of F'. The construction is recursive. First, at the lowest level, $f_{-1} : \mathbb{Z} \to \mathbb{Z}$ is simply the identity map. Next, we assume that the map f_{k-1} has been constructed, and proceed to construct f_k. We choose a $\mathbb{Z}G$-basis of F_k, $e_{k,i}$. Eq. (15) demands that f_k satisfies

$$\partial'_k f_k(e_{k,i}) = f_{k-1}(\partial_k e_{k,i}).$$

It is straightforward to check that the r.h.s is closed. Hence, as discussed in Sec. II A, Eq. (6) implies that we can choose the image of $e_{k,i}$ to be

$$f_k(e_{k,i}) = s'_{k-1} f_{k-1}(\partial_k e_{k,i}). \tag{16}$$

We then extend f_k linearly to F_k.

We notice that, even with a given s', the chain map f constructed above is not unique. It depends on the choice of the basis in each F_k, because the contracting homotopy s' does not commute with the G-action. However, different choices of f are homotopically equivalent to each other, as we shall see explicitly in Sec. III B.

Actually, in the above construction, only the contracting homotopy s' of the second resolution F' is used. Therefore, the chain map can be constructed from an arbitrary chain complex F made of free-G- modules, even if F is not contractible.

Using a chain map $f : F \to F'$, one can map a cocycle in the basis of F' to one in the basis of F, using the pullback map f^*. For a cochain $\alpha' \in \mathrm{Hom}_G(F', M)$, its image $f^*(\alpha)$ is given by the following relation,

$$\forall x \in F, \langle f^*(\alpha), x\rangle = \langle \alpha, f(x)\rangle. \tag{17}$$

In particular, in this work, we usually consider chain maps between two types of resolutions of G: F is a reduced resolution given by the algorithm in HAP, and $\bar{F}$ is the normalized bar resolution discussed in Sec. II A. We denote the two chain maps between them by $f : F \to \bar{F}$ and $g : \bar{F} \to F$, respectively. Since both F and $\bar{F}$ have explicit contracting homotopies, both f and g can be constructed recursively using the algorithm in Eq. (16).

We end this section with an example of computing the chain maps. Again, we consider the finite cyclic group $G = \mathbb{Z}_n$. Its reduced resolution F, derived from the chain complex of the lens space, is given in Sec. II A, along with a contracting homotopy.

We now demonstrate the construction of $f : F \to \bar{F}$. First, since both F_0 and $\bar{F}_0$ are simply $\mathbb{Z}G$ with one basis, f_0 just maps the basis $e_0 \in F_0$ to the basis $[\cdot] \in \bar{F}_0$. (Recall that basis in F_n are labeled by n group elements. Hence, the single basis of F_0 is labeled by zero group element, and denoted by $[\cdot]$.) Next, we use Eq. (16) to construct f_1:

$$f_1(e_1) = s_0 f_0(\partial e_1) = s_0(a[\cdot] - [\cdot]) = [a]. \tag{18}$$

Similarly, we can proceed and compute f_2 and f_3 recursively,

$$f_2(e_2) = [a|a] + [a^2|a] + \cdots + [a^{n-1}|a], \tag{19}$$

$$f_3(e_3) = [a|a|a] + [a|a^2|a] + \cdots + [a|a^{n-1}|a], \tag{20}$$

$$f_4(e_4) = \sum_{i,j=1}^{n-1} [a^i|a|a^j|a]. \tag{21}$$

Next, we demonstrate constructing $g : \bar{F} \to F$. Comparing to f, the results are more lengthy. Hence, we only compute the first two dimensions, which are used in the example of the main text. Similar to f_0, g_0 also maps the single basis $[\cdot]$ in $\bar{F}_0$ to the single basis e_0 in F_0. Next, we compute g_1:

$$g_1([a^i]) = s_0 g_0(\partial[a^i]) = s_0(a^i e_0 - e_0) = (1+\cdots+a^{i-1})e_1. \tag{22}$$

In the last step, we used the contracting homotopy of the resolution in Eq. (14). Next, we compute g_2:

$$\begin{aligned} g_2([a^i|a^j]) &= s_1 g_1(\partial[a^i|a^j]) = s_1 g_1(a^i[a^j] - [a^{i+j}] + [a^i]) \\ &= s_1\left\{(1+\cdots+a^{i+j-1})e^1 - (1+\cdots+a^{l-1})e^1\right\}, \end{aligned} \tag{23}$$

where $l = i + j \mod n$. Hence, if $i + j < n$, we have $l = i + j$, andthe above equation vanishes. If $n \leq i + j < 2n$, we have $l = i + j - n$, and the above equation gives

$$g_2([a^i|a^j]) = s_1\left\{(a^{l-1} + \cdots + a^{l+n-1})e^1\right\} = e^2.$$

Combing these two cases, we have

$$g_2([a^i|a^j]) = \left\lfloor \frac{i+j}{n} \right\rfloor e^2, \tag{24}$$

where $\lfloor (i+j)/n \rfloor$, meaning the greatest integer less than or equal to $(i+j)/n$, is 0 (1) if $i+j < (\geq)n$, respectively.

C. Coboundary check

As we discussed in Sec. I, the most time-consuming task of computing an SPT classification is to check whether a obstruction function, which is a cocycle, is a trivial coboundary or not. Such an obstruction cocycle is often expressed as an inhomogeneous cocycle. Checking whether a cocycle is a coboundary using the normalized bar resolution is quite time-consuming, since the size of the coboundary matrix is $(|G|-1)^n$ by $(|G|-1)^{n+1}$. In contrast, performing the coboundary check is much easier using the reduced resolution F, because the dimensions of the G modules F_n and F_{n+1} are much smaller.

Hence, we propose the following approach for checking whether an inhomogeneous cocycle $\bar{\alpha} \in \bar{Z}^n(G, M)$ is a coboundary. First, we construct a reduced resolution F, and the chain map $f : F \to \bar{F}$. Second, we map $\bar{\alpha}$ to a cocycle $\alpha \in \mathrm{Hom}_G(F, M)$, using the pullback map, as $\alpha = f^*(\bar{\alpha})$. Finally, we check whether the cocycle α is trivial, using the reduced resolution F.

To be more concrete, this approach can be implemented using the following algorithm. To check the trivialness of a n-cocycle α, we use the Smith normal form of the coboundary map $d^{n-1} : \mathrm{Hom}_G(F_{n-1}, M) \to \mathrm{Hom}_G(F_n, M)$. The Smith normal form reveals a set of invariants identifying nontrivial cocycles:

$$I_k = \sum_i a_{k,i} \langle \alpha, e_{n,i} \rangle. \tag{25}$$

A nonvanishing $I_k \neq 0$ indicates that the cocycle α is not a coboundary. The details of obtaining these invariants from the Smith normal form of d^{n-1} are reviewed in Sec. VII. Next, we express the invariants I_k with $\bar{\alpha}$, using the chain map f. Using Eq. (17), we write $\alpha(e_{n,i})$ as $\bar{\alpha}(f(e_{n,i}))$, and the invariants in Eq. (25) as

$$I_k = \sum_i a_{k,i} \langle \bar{\alpha}, f(e_{n,i}) \rangle. \tag{26}$$

Finally, we compute each I_k using the entries of $\bar{\alpha}$, and check if all I_k vanish. Any nonvanishing I_k indicates that $\bar{\alpha}$ is a nontrivial cocycle. Since F is usually much smaller than $\bar{F}$ (to be more precise, the dimensions of F_n, $\mathrm{rank}_{\mathbb{Z}G} F_n$, are much smaller than that of $\bar{F}_n$), this algorithm can save significant computational costs comparing to the naive approach using only the inhomogeneous cocycles.

We will demonstrate this algorithm and the saving on computational costs using the example of checking a 3-cycle for a cyclic group $G = \mathbb{Z}_n$. As we see in Sec. II A, the modules F_k only have one $\mathbb{Z}G$-basis e_k. The boundary operator $\partial : F_3 \to F_2$ is given by $\partial e_3 = (x-1)e_2$. Hence, the corresponding coboundary operator is simply a one-by-one matrix. If the coefficient module is $M = \mathrm{U}(1)$ with a trivial G-action, the coboundary operator $d^2 : \mathrm{Hom}_G[F_2, \mathrm{U}(1)] \to \mathrm{Hom}_G[F_3, \mathrm{U}(1)]$ vanishes: $d^2 = 0$. Hence, there is no nontrivial coboundary equivalence, and any cocycle with a nonvanishing entry $\alpha(e_3) \neq 0$ is a nontrivial cocycle. In other words, to check the trivialness of a cocycle, we need to examine one invariant $I_1 = \langle \alpha, e_3 \rangle$. Using the chain map in Eq. (20), we express this invariant in terms of the inhomogeneous cocycle $\bar{\alpha}$,

$$I_1 = \sum_{j=1}^{n-1} \bar{\alpha}(a, a^j, a). \tag{27}$$

This is directly related to the partition function in Eq. (2): the partition function is $Z = e^{2\pi i I_1}$. This demonstrates that computing the invariants in Eq. (26) is equivalent to computing the partition functions on the representative G-bundles discussed in Sec. I.

III. APPLICATION TO SPT CLASSIFICATION

A. Computing obstruction function

We combine the algorithms introduced in Sec. II to compute the obstruction functions that appear in fSPT classification.

As an example, we discuss the obstruction function $O_5[n_2]$ in the classification of (3+1)D fSPT, which maps a Majorana decoration pattern, represented by a 2-cocycle $n_2 \in H^2[G_b, \mathbb{Z}_2]$, to an obstruction class represented by a 5-cocycle in $H^5[G_b, \mathrm{U}(1)_T]$. Here, we consider the simple case, where the total symmetry group G_f is a direct product of the bosonic symmetry group G_b and the fermion-parity symmetry $\mathbb{Z}_2^f$. The more general cases where G_f is a nontrivial group extension of G_b over $\mathbb{Z}_2^f$ can be computed in a similar manner.

In terms of inhomogeneous cochains, the obstruction function is constructed in the following steps [8]: First, one computes the $O_4[n_2]$ obstruction function, given by the following formula,

$$O_4[n_2] = n_2 \cup n_2. \tag{28}$$

The cup product in this equation is defined in Sec. VIII. We then check whether the obstruction $O_4[n_2]$ vanishes, meaning that it is a coboundary. This is because if it is a nontrivial cocycle, such n_2 will lead to violation of fermion-parity conservation and does not represent consistent Majorana-chain decorations in a 3D fSPT state. Second, if $O_4[n_2]$ is a trivial coboundary, we need to find a solution of the equation

$$dn_3 = O_4[n_2] = n_2 \cup n_2. \tag{29}$$

Third, using the solution n_3, one can compute the obstruction O_5, Eq. (220) in [8], One then needs to check if the computed O_5 is a trivial cocycle.

Although to our best knowledge, the obstruction function O_5 can only be expressed using inhomogeneous cocycle, this calculation can be accelerated using the reduced resolution and the algorithms presented in previous sections. First, we enumerate all cohomology classes n_2 in $H^2[G, \mathbb{Z}_2]$ using cochains in the reduced resolution. Next, we map n_2 to an inhomogeneous cochain, $\bar{n}_2 = g^* n_2$. This allows us to compute $O_4[\bar{n}_2]$ using the cup-product formula in Sec. VIII directly. We then check whether it is a trivial obstruction class using the algorithm in Sec. II C. (In this step, the cup product can also be computed directly in the reduced resolution, by constructing a diagonal approximation using the contracting homotopy, as described in Sec. VIII.) If $O_4[\bar{n}_2]$ is trivial, we can construct a solution of Eq. (29) using the algorithm in Sec. III B. We then compute O_5 and check its trivialness using the algorithm in Sec. II C.

The computational cost can be further reduced using lazy evaluation, which is a commonly used method in programming and can be easily implemented in modern programming languages. We demonstrate the use of lazy evaluation using the example of O_5 and $G = \mathbb{Z}_n$. Naively, to check if O_5 is trivial, one first computes $\bar{\alpha} = O_5$ and then check its trivialness. Since there are $(|G|-1)^5 = (n-1)^5$ entries of $\bar{\alpha}$, the cost of this step scales as $(n-1)^5$. However, using the algorithm in Sec. II C, one only needs to check that all topological invariants I_m vanish. Following the steps in Sec. II B, one finds that there is only one invariant, given by

$$I = \sum_{1<i,j<n} \bar{\alpha}(a, a^i, a, a^j, a). \tag{30}$$

This invariant envolves only $(n-1)^2$ entries of $\bar{\alpha}$. Therefore, only these entries need to be computed from Eq. (220) in [8] Skipping the rest of the entries reduces the computational cost from $O[(n-1)^5]$ to $O[(n-1)^2]$. In practice, one only passes the functional form of $\bar{\alpha}$ given by Eq. (220) in [8] instead of all its entries, to the trivialness-checking procedure. This procedure then constructs the invariants and computes the cochain entries on the fly when they are needed. This practice of deferring the evaluation of the entries until their values are needed is called lazy evaluation in programming. In this way, both CPU and memory costs are saved.

B. Solving cocycle equations

The reduced resolution can also be used to accelerate the task of finding one solution of the cocycle equation,

$$d\bar{\beta} = \bar{\alpha}, \tag{31}$$

where α is a $(k+1)$-coboundary (otherwise this equation has no solution). This task can also be time-consuming using the inhomogeneous cocycles, as the matix form of Eq. (31) has dimension $(|G|-1)^k \times (|G|-1)^{k+1}$.

Naively, one may try to solve Eq. (31) by mapping $\bar{\alpha}$ to a cochain in the reduced resolution using the pullback of the chain map $f: F \to \bar{F}$ as $\alpha = f^*\bar{\alpha}$, find a solution $\beta = d\alpha$ there, and map it back to an inhomogeneous cocycle as $g^*\beta$, using $g: \bar{F} \to F$. However, $g^*\beta$ constructed this way is not a solution of Eq. (31), because the two chain maps f and g are not the inverse of each other. In fact, the composition fg cannot be the identity map, because the modules $\bar{F}_k$ have higher dimensions than F_k. Instead, fg is only homotopic to the identity map, meaning that it can be related to identity using a homotopy $h: fg \sim \mathrm{id}$.

A homotopy h is a degree-1 map: $h_k: \bar{F}_k \to \bar{F}_{k+1}$, illustrated by the following diagram,

$$\begin{tikzcd}
\cdots \arrow[r] & \bar{F}_{k+1} \arrow[r, "\partial_{k+1}"] \arrow[d] & \bar{F}_k \arrow[r, "\partial_k"] \arrow[d] \arrow[dl, "h_k"] & \bar{F}_{k-1} \arrow[r] \arrow[d] \arrow[dl, "h_{k-1}"] & \cdots \\
\cdots \arrow[r] & \bar{F}_{k+1} \arrow[r, "\partial_{k+1}"] & \bar{F}_k \arrow[r, "\partial_k"] & \bar{F}_{k-1} \arrow[r] & \cdots
\end{tikzcd} \tag{32}$$

Here, the verticle arrows represent the difference between fg and identity, $f_k g_k - \mathrm{id}_{\bar{F}_k}$. The diagram is not a commutative diagram: it instead satisfies

$$\partial_{k+1} h_k + h_{k-1}\partial_k = f_k g_k - \mathrm{id}_{\bar{F}_k}. \tag{33}$$

For given chain maps f and g, the homotopy h can also be constructed recursively using the contracting homotopy s of F, in the following way similar to the algorithm in Sec. II B.

Similar to the construction in Sec. II B, this is done recursively. For simplicity, we assume that both F_0 and $\bar{F}_0$ has only one $\mathbb{Z}G$ basis. Therefore, f_0 and g_0 simply maps between the two unique basis, and consequently $f_0 g_0$ is exactly the identity map. As a result, we can choose $h_0 = 0$ because there is nothing to correct. This is the starting point of our construction. Next, we assume that h_{k-1} has been constructed, and proceed to construct h_k. We take a $\mathbb{Z}G$-basis e_i of $\bar{F}_k$, and the property (33) demands that

$$\partial h_k(e_i) = -h_{k-1}\partial e_i + f_k g_k(e_i) - e_i. \tag{34}$$

We notice that the r.h.s. of this equation can be computed from existing constructions. A solution of Eq. (34) can be found using the contracting homotopy $\bar{s}$ of the resolution $\bar{F}$,

$$h_k(e_i) = \bar{s}_k[-h_{k-1}\partial e_i + f_k g_k(e_i) - e_i]. \tag{35}$$

We can then extend h_k linearly to $\bar{F}_k$.

The homotopy h can be used to construct a solution of Eq. (31), as it corrects the difference between fg and identity. Since it is a degree-1 map $h_k: \bar{F}_k \to \bar{F}_{k+1}$, its pullback maps a $(k+1)$-cochain $\bar{\alpha}$ to a k-cochain $h^*\bar{\alpha}$. We can use it to augment $g^*\beta$ and construct a solution as

$$\bar{\beta} = g^*\beta - h^*\bar{\alpha}. \tag{36}$$

We now prove that this is indeed a solution of Eq. (31). Consider any $x \in \bar{F}_{k+1}$. Using the definition of the pullback maps, we get

$$\langle d\bar{\beta}, x\rangle = \langle d\beta, g(x)\rangle - \langle \bar{\alpha}, h(\partial x)\rangle. \tag{37}$$

Since $d\beta = \alpha$, we get

$$\langle d\bar{\beta}, x\rangle = \langle \alpha, g(x)\rangle - \langle \bar{\alpha}, h\circ\partial(x)\rangle = \langle \bar{\alpha}, f\circ g(x)\rangle - \langle \bar{\alpha}, h\circ\partial(x)\rangle. \tag{38}$$

Using Eq. (33), we get

$$\langle d\bar{\beta}, x\rangle = \langle \bar{\alpha}, x\rangle + \langle \bar{\alpha}, \partial\circ h(x)\rangle. \tag{39}$$

Since $d\bar{\alpha} = 0$, the second term in r.h.s. vanishes. Hence, we conclude that $d\bar{\beta} = \bar{\alpha}$.

IV. PHYSICAL INTERPRETATION

In Sec. II, we describe an algebraic algorithm that generates the topological invariants differentiating SPT phases. The invariants generated by the algorithm coincide with the partition functions evaluated on handpicked representative G-bundles. In this section, we give an interpretation of the connection between the two. For simplicity, we assume that G is a finite unitary symmetry group, and thus consider the group cohomology with U(1) coefficients. The results can be easily generalized to include antiunitary symmetry operations, and infinite groups.

We first review the connection between SPT states and group cohomology computed from an arbitrary resolution. A k-cocycle $\alpha \in \mathrm{Hom}_G[F_k, \mathrm{U}(1)]$, which is a cocycle in $H^k[BG, \mathrm{U}(1)]$, can be viewed as an action, mapping each k-cell $\sigma \in BG_k$ to a U(1) phase factor $\langle \alpha, \sigma \rangle$. Because EG is a universal bundle, such an action can be used to construct a partition function for any G-bundle over a k-dimensional orientable space-time manifold B [2]. For simplicity, we also assume B is connected. Because G is a finite discrete group, the gauge connection on B must vanish. Therefore, the G-bundle is specified by the symmetry flux through each noncontractible loop in B, which can be expressed as a group homomorphism $\gamma : \pi_1(B) \to G$. Since $\pi_1(BG) = G$, γ is also a homomorphism $\gamma : \pi_1(B) \to \pi_1(BG)$. Because all higher homotopy groups of BG vanish, γ can be further uniquely (up to homotopy) extended to a cellular map $\gamma : B \to BG$. Algebraically, the cellular map γ is a chain map from the chain complex of B to that of BG, which is the resolution F. Such a chain map can be constructed using the algorithm in Sec. II B, using a contracting homotopy of F. In particular, γ maps each k-cell $\sigma \in B_k$ to an algebraic sum of k-cells in BG, denoted by $\gamma(\sigma)$. Intuitively, this can be viewed as a decomposition of $\sigma \in B_k$ using cells in BG. One can then evaluate α on each cell in $\gamma(\sigma)$, and define the sum of the evaluations as the value of the action on σ. Mathematically, this is expressed as $\langle \gamma^* \alpha, \sigma \rangle = \langle \alpha, \gamma(\sigma) \rangle$, where $\gamma^*(\alpha)$ is the pullback of α by γ, which is a cochain on B, and can be viewed as an action induced by α defined on B. Finally, one can integrate $\gamma^* \alpha$ on B, and construct the following partition function,

$$Z = \exp(2\pi i \langle \gamma^* \alpha, [B] \rangle). \tag{40}$$

Here, $[B]$ denotes the fundamental class of B [3], which is an algebraic sum of all k-cells of B, with signs given by comparing the orientation of each cell to a global orientation of B.

In particular, when we take F to be the bar resolution, $\gamma : B \to BG$ can be viewed as a simplicial decomposition, or a triangulation, of B: the image $\gamma([B])$ gives an algebraic sum of all simplices in such a decomposition,

$$\gamma([B]) = \sum s_{i_0 \cdots i_k} [v_{i_0} \cdots v_{i_k}].$$

Hence, Eq. (40) becomes the following function,

$$Z = \exp\left\{ 2\pi i \sum_{[v_{i_0} \cdots v_{i_k}]} s_{i_0 \cdots i_k} \bar{\alpha}(g_{i_0 i_1}, \ldots, g_{i_{k-1} i_k}) \right\}, \tag{41}$$

where g_{ij} is the gauge connection on the 1-cell ij. For a cocycle $\bar{\alpha}$, the above partition function is independent of the choices of g_{ij}, as long as the total symmetry flux $\prod g_{ij}$ along noncontractible loops stays the same. Therefore, one can sum over all possible choices of g_{ij} and obtain the familiar form of SPT-state partition function,

$$Z = \sum_{g_{ij}} \exp\left\{ 2\pi i \sum_{[v_{i_0} \cdots v_{i_k}]} s_{i_0 \cdots i_k} \bar{\alpha}(g_{i_0 i_1}, \ldots, g_{i_{k-1} i_k}) \right\}. \tag{42}$$

Next, we notice that the invariants I_l introduced in Sec. II C can be viewed as partition functions of representative G-bundles. Each bundle is based on a k-dimensional space-time manifold B_l, with a decomposition $\gamma_l : B \to BG$. In particular, the fundamental class maps to $\gamma_l([B_l]) = \sum_i a_{l,i} e_{n,i}$. The invariant I_l is then given by

$$I_l = \langle \gamma_l^* \alpha, [B_l] \rangle. \tag{43}$$

Using Eq. (40), we see that the partition function Z is given by $\exp(I_l)$.

As an example, we revisit the representative G-bundle studied in Sec. I. In fact, the manifold in Fig. 1(a) can be viewed as a CW-complex with one 3-cell, one 2-cell, one 1-cell and one 0-cell, respectively (see the caption of the figure). Since the manifold is the three-dimensional lens space $L_3(1)$ and BG is the infinite-dimensional lens space $L_3(1, 1, 1, \ldots)$, the chain map $\gamma : B \to BG$ maps each k-cell to the single k-cell in BG, which corresponds to the single generator of F_k discussed in Sec. II A. In particular, the fundamental class of B is mapped to e_3. Hence, the partition function in Eq. (40) is given by $Z = e^{2\pi i I_1}$, where $I_1 = \langle \alpha, e_3 \rangle$.

Finally, the chain map $f : F \to \bar{F}$ can be viewed as a simplicial decomposition, or a triangulation, of cells in the CW-complex BG, whose chain complex is given by F. In fact, f can be viewed as a special case of celluar maps between a G-bundle and a classifying space of G, as $f : BG \to \overline{BG}$. Here, BG and $\overline{BG}$ denote a CW-complex and a simplicial complex, respectively, both serving as classifying spaces of G, and their chain complexes are given by F and $\bar{F}$, respectively. Consequently, the composition $f \circ \gamma_l : B \to \overline{BG}$ gives a triangulation of the manifold B_l, and the partition function

$$Z = \exp\{2\pi i \langle (f \circ \gamma_l)^* \alpha, [B_l] \rangle\} = \exp\{2\pi i \langle (f^* \alpha, \gamma_l([B_l]) \rangle\} \tag{44}$$

then computes the partition function on B_l using the inhomogeneous cocycle $\bar{\alpha} = f^* \alpha$.

Combining the above understanding, we see that the invariants computed in Sec. II C are the partition function of the representative G-bundles, computed from inhomogeneous cocycles using a triangulation. In particular, the triangulation is constructed algebraically using the chain map $f: F \to \bar{F}$. Such constructions are performed automatically by the algorithm in Sec. II.

V. INHOMOGENEOUS COCYCLES

In this section, we review the inhomogeneous cocycles, a tool widely used to compute the cohomology of finite groups and to construct SPT classification.

An inhomogeneous n-cochain $\alpha \in C^n(G, M)$ is a function mapping n group elements $g_1, \ldots, g_n$ to a coefficient $\alpha(g_1, \ldots, g_n) \in M$. Here, M is a $\mathbb{Z}G$ module, refered to as the coefficient module of the group cohomology. The most common coefficient we encounter in SPT classification is the $\mathrm{U}(1)_T$ module: In our notation, the U(1) module is actually a real number modulo one, which is often denoted as $\mathbb{R}/\mathbb{Z}$. Physically, it represents a U(1) phase factor. The subscript T denotes how the symmetry group G acts on this module: $g \cdot \phi = -\phi$ if g is an antiunitary operation, like the time-reversal symmetry T.

The coboundary operator $d^n : C^n(G, M) \to C^{n+1}(G, M)$ maps a n-cochain to a $(n+1)$-cochain, and it is defined as the following,

$$\begin{aligned}(d\alpha)(g_1, \ldots, g_{n+1}) =& g_1\alpha(g_2, \ldots, g_{n+1}) - \alpha(g_1g_2, g_3, \ldots, g_{n+1}) + \alpha(g_1, g_2g_3, \ldots, g_{n+1}) \\ &+ \cdots + (-1)^n\alpha(g_1, \ldots, g_ng_{n+1}) + (-1)^{n+1}\alpha(g_1, g_2, \ldots, g_n).\end{aligned} \tag{45}$$

VI. THE INTEGRAL GROUP RING

In this section, we briefly review the basic concepts of the integral group ring $\mathbb{Z}G$, and a free $\mathbb{Z}G$ module.

For any group G, we construct the integral group ring $\mathbb{Z}G$ as follows. The elements of $\mathbb{Z}G$ are formally linear combination of group elements with integral coefficients, $x = \sum_{g\in G} x_g g$, $x_g \in \mathbb{Z}$. The addition and multiplication between two elements $x = \sum_{g\in G} x_g g$ and $y = \sum_{g\in G} y_g g$ are given by

$$x + y = \sum_{g\in G}(x_g + y_g)g,$$

$$xy = \sum_{g,h\in G} x_g y_h (gh),$$

respectively. It is straightforward to check that $\mathbb{Z}G$ is a ring.

For an arbitrary ring R, a free R-module M can be understood as an analog of a linear space, with coefficients in R instead. The module M can be generated by a R-basis, denoted by e_i.

VII. SMITH NORMAL FORM

In this section, we briefly review the Smith normal form (SNF) of an integral matrix, which is used in Sec. II C.

We consider an $n \times m$ matrix over $\mathbb{Z}$: $A = A_{ij}$. Its SNF is a decomposition into three matrices, L, R and Λ, such that

$$LAR = \Lambda, \tag{46}$$

where L and R are $n\times n$ and $m\times m$ unimodular matrices, respectively, and Λ is a diagonal matrix of dimensions $n \times m$. As unimodular matrices, the inverse matrices of L and R are also integral matrices.

As an application of the SNF, we consider the coboundary condition $\alpha = d\beta$. Here, d is the coboundary map $d^{n-1} : \mathrm{Hom}_G[F_{n-1}, \mathrm{U}(1)] \to \mathrm{Hom}_G[F_n, \mathrm{U}(1)]$. Assume that $\mathrm{rank}_{\mathbb{Z}G} F_{n-1} = m$ and $\mathrm{rank}_{\mathbb{Z}G} F_n = n$, respectively, and denote a set of $\mathbb{Z}G$ basis of F_{n-1} and F_n by e_i^n and e_i^{n-1}, respectively. A cochain α in $\mathrm{Hom}_G[F_n, \mathrm{U}(1)]$ is represented as a vector α_i using its components on e_i^n, $\alpha_i = \langle \alpha, e_i^n\rangle$. Similarly, a cochain β in $\mathrm{Hom}_G[F_{n-1}, \mathrm{U}(1)]$ is represented as a vector β_i using its components on e_i^{n-1}, $\beta_i = \langle \beta, e_i^{n-1}\rangle$. The coboundary map $\alpha = d^{n-1}\beta$ can then be represented as a matrix A^i_j, such that

$$\alpha_i = \sum_{j=1}^{m} A_{ij}\beta_j, \tag{47}$$

According to Eq. (8), the explicit form of A^i_j can be obtained by expanding ∂e_i^n on basis of e_j^{n-1},

$$\partial e_i^n = \sum_{j=1}^{m} A_{ij} e_j^{n-1}. \tag{48}$$

Here, the group-element coefficients are converted to numbers using the group action on the coefficients.

To solve Eq. (47) and check if α is a coboundaries, we find the SNF of matrix A given by Eq. (46). As a result, Eq. (47) is changed into

$$L\alpha = \Lambda\beta', \tag{49}$$

where $\beta' = R^{-1}\beta$. Since R is unimodular, going through all possible β is equivalent to going through all possible β'. Hence, α is a coboundary if and only if there is a β' such that Eq. (49) holds. Therefore, we consider each row in the matrix equation (49), which has the following form,

$$\sum_{j=1}^{n} L_{ij}\alpha_j = \Lambda_{ii}\beta_i'. \tag{50}$$

For each diagonal element $\Lambda_{ii} = 0$, the LHS of Eq. (50) defines an invariant, which we denote by I^j,

$$I_i = \sum_{j=1}^{m} L_{ij}\alpha_j. \tag{51}$$

Since $\Lambda_{ii} = 0$, Eq. (50) implies that a coboundary must satisfy $I_i = 0$. Therefore, a nonvanishing $I_i \neq 0$ indicates that α is not a coboundary.

VIII. CUP AND HIGHER-CUP PRODUCTS

In this section, we briefly review the concept of cup products and higher-cup products in group-cohomology theory, which appears frequently in formulas computing the classification of fSPT and SET phases.

The cup product is a group-cohomology operation that maps a pair of cocycles to another cocycle. The mathematical definition of cup products can be found in Chap. V of Ref. [5]. In particular, it maps a p-cocycle and a q-cocycle to a $(p+q)$-cocycle:

$$\cup : H^p(G, M_1) \times H^q(G, M_2) \to H^{p+q}(G, M_3). \tag{52}$$

Here, in general, M_1, M_2 and M_3 are three different G-modules, with a bilinear form $B : M_1 \times M_2 \to M_3$.

As we discuss in Sec. II A, cocycles in a cohomology group can be expressed using different resolutions of G. Using the inhomogeneous cocycles, the cup product is given by the following explicit form,

$$\alpha \cup \beta(g_1, \ldots, g_{p+q}) = B[\alpha(g_1, \ldots, g_p), g_1 \cdots g_p \cdot \beta(g_{p+1}, \ldots g_{p+q})]. \tag{53}$$

In fact, this definition provides a cup product on the inhomogeneous cochains,

$$\cup : C^p(G, M_1) \times C^q(G, M_2) \to C^{p+q}(G, M_3). \tag{54}$$

The $\cup$ product satisfies the well-known Leibniz formula,

$$d(\alpha \cup \beta) = d\alpha \cup \beta + (-1)^{\deg \alpha}\alpha \cup d\beta. \tag{55}$$

This implies that if both α and β are cocycles, $\alpha \cup \beta$ is also a cocycle, which is consistent with Eq. (52).

On an arbitrary resolution F over $\mathbb{Z}G$, the definition of a cup product is not as straightforward as Eq. (53). In general, it requires constructing a so-called diagonal approximation Δ, which is a chain map $\Delta : F \to F \otimes F$. Here, $F \otimes F$ is the tensor product of F with itself, with a diagonal G-action. Using such a diagonal approximation, one can define a cup product similar to Eq. (54):

$$\cup : \mathrm{Hom}_G(F_p, M_1) \times \mathrm{Hom}_G(F_q, M_2) \to \mathrm{Hom}_G(F_{p+q}, M_3). \tag{56}$$

(See Chap V of Ref. [5] for the details of the diagonal approximation and cup products.) In particular, the cup product defined using a diagonal approximation also satisfies Eq. (55). As a result, it also gives a cup product between cohomology classes as in Eq. (52). It is important to notice that the cup product defined in Eq. (56) is not unique, as there are many possible choices of the diagonal approximation. However, different choices of F and diagonal approximations always lead to the same cup product between cohomology classes in Eq. (52).

In practice, for an arbitrary resolution R, a cup product can be constructed in the following steps: First, construct the tensor product $F \otimes F$ with a diagonal G-action. Second, construct a chain map $\Delta : F \to F \otimes F$, which serves as a diagonal approximation, using the algorithm in Sec. II B. Last, a cup product is constructed using this diagonal approximation. These steps can construct a cup product without the help of inhomogeneous cocycles. Alternatively, using the ideas in Sec. III A, one can compute the cup product by first mapping the cocycles to inhomogeneous cocycles, computing the cup product using Eq. (53), and then mapping the result back. In general, we expect the first approach to be more efficient, because it skips the intermediate steps involving inhomogeneous cocycles. However, in practice, we choose to use the second approach. This is because there are usually more complicated obstruction functions that cannot be written entirely in terms of cup products (and higher cup products), which takes much longer to compute and can only be computed using the method in Sec. III A. Therefore, the computational cost is not a big issue here. Consequently, we choose the second approach because it has a uniform realization with other obstruction functions.

The higher cup products can be defined in a similar way. First, for inhomogeneous cocycles, there are explicit definitions of the higher cup products, which can be found in Ref. [6]. A cup-k product maps a p-cochain and a q-cochain to a $(p+q-r)$-cochain,

$$\cup_k : C^p(G, M_1) \times C^q(G, M_2) \to C^{p+q-k}(G, M_3). \tag{57}$$

In particular, $\cup_0$ is nothing but the cup product defined above. Below, we give the explicit form of $\cup_1$ and $\cup_2$, which were used in obstruction functions for fSPTs.

$$(\alpha \cup_1 \beta)(g_1, \ldots, g_{p+q-1}) = \sum_{i=0}^{p-1} (-1)^{(p-i)(q+1)} B[\alpha(g_1, \ldots, g_i, g_{i+1} \cdots g_{i+q}, g_{i+q+1}, \ldots, g_{p+q-1}), g_1 \cdots g_i \beta(g_{i+1}, \ldots, g_{i+q})]. \tag{58}$$

$$(\alpha \cup_2 \beta)(g_1, \ldots, g_{p+q-2}) = \sum_{0 \le i < j \le p} (-1)^{(p-i)(j-i+1)} B[\alpha(g_1, \ldots, g_i, g_{i+1} \cdots g_j, g_{j+1}, \ldots, g_{j-i+p}), g_1 \cdots g_i \beta(g_{i+1}, \ldots, g_j, g_{j+1} \cdots g_{j-i+p}, g_{j-i+p+1}, \ldots g_{p+q-2})]. \tag{59}$$

The cup-k product satisfies the following relation (Thm. 5.1 of Ref. [6]),

$$d(\alpha \cup_k \beta) = (-1)^{p+q-k} \alpha \cup_{k-1} \beta + (-1)^{pq+p+q} \beta \cup_{k-1} \alpha + d\alpha \cup_k \beta + (-1)^p \alpha \cup_k d\beta. \tag{60}$$

As a result, the $\cup_k$ product gives a product between cohomology classes,

$$\cup_k : H^p(G, M_1) \times H^q(g, M_2) \to H^{p+q-k}(G, M_3). \tag{61}$$

The higher cup products can also be constructed on an arbitrary resolution, without using the inhomogeneous cocycles. This is done using the higher diagonal approximations [7]. The higher diagonal approximations are series of homotopy equivalences, which can be constructed recursively using the method in Sec. III B. This allows us to compute higher cup products without going through the inhomogeneous cocycles. However, in practice, we choose to use the approach of mapping to/from the inhomogeneous cocycles, for similar reasons as in the case of the cup product.

IX. THE SPTSET PACKAGE

The algorithm described in this work is implemented in the package SptSet for the GAP software. It can be used to compute the classification of fSPT states protected by 2D wallpaper groups, which is listed in the main text. Once the package is installed following the instruction on its website, the results can be computed by running the script in examples/fspt_2d_ez.g and examples/fspt_2d_s12.g, respectively. The full design and functionality of this package will be reported elsewhere.

[1] N. Tantivasadakarn, Phys. Rev. B **96**, 195101 (2017).
[2] R. Dijkgraaf and E. Witten, Comm. Math. Phys. **129**, 393 (1990).
[3] A. Hatcher, *Algebraic topology* (Cambridge University Press, 2002).
[4] D. Joyner, (2007), arXiv:0706.0549 [math.GR].
[5] K. S. Brown, *Cohomology of groups*, Vol. 87 (Springer Science & Business Media, 2012).
[6] N. E. Steenrod, Annals of Mathematics **48**, 290 (1947).
[7] J. F. Davis, in *Algebraic and Geometric Topology*, edited by A. Ranicki, N. Levitt, and F. Quinn (Springer Berlin Heidelberg, Berlin, Heidelberg, 1985) pp. 51–61.
[8] Q.-R. Wang and Z.-C. Gu, Phys. Rev. X **10**, 031055 (2020).
[9] G. Brumfiel and J. Morgan, (2018), arXiv:1803.08147 [math.GT].