

Precision estimates of large charge RG exponents Y_q in the 3D XY universality class

Martin Hasenbusch 

Institut für Theoretische Physik, Universität Heidelberg, Philosophenweg 19, 69120 Heidelberg, Germany



(Received 30 November 2025; accepted 7 January 2026; published 26 January 2026)

We accurately compute the renormalization group (RG) exponents Y_q of large q fields at the $O(2)$ invariant fixed point in three dimensions. We build on an iterative approach that has been previously proposed and is implemented by using the worm algorithm. We simulate an improved XY model, that has next-to-next-to-nearest couplings in addition to nearest ones. In the worm update we incorporate weights, which allows us to obtain accurate results up to $q = 64$. For example we get $Y_q = 1.76370(12)$, $0.89167(23)$, and $-0.11203(34)$ for $q = 2, 3$, and 4 , respectively. The comparison with the large q effective field theory gives an excellent agreement down to $q = 4$ and provides accurate estimates of the parameters of the effective field theory.

DOI: [10.1103/bnlx-8q4q](https://doi.org/10.1103/bnlx-8q4q)

I. INTRODUCTION

Critical phenomena [1–5] in three dimensions have been studied by using various theoretical approaches such as field theoretic methods, series expansions and Monte Carlo (MC) simulations of lattice models and more recently the conformal bootstrap (CB) method [6,7]. The theoretical framework is the renormalization group (RG), which is the basis of various methods, and provides us with fundamental notions such as universality. It has been realized that in the large spin and large charge limit, semiclassical approximations are viable. See for example Refs. [8–12]. For a review see [13]. Here we are concerned with large charge q in the simplest case, the XY-universality class in three dimensions. The dimension D_q of the lowest operator with global charge q behaves as [11–13]

$$D_q = c_{3/2}q^{3/2} + c_{1/2}q^{1/2} - 0.0937256 + O(q^{-1/2}). \quad (1)$$

Here we denote the dimension of the operator by D_q , since later we shall use $\Delta_q = D_q - D_{q-1}$. The corresponding RG exponent is $Y_q = d - D_q$, where here $d = 3$ is the spatial dimension of the system. The coefficients $c_{3/2}$ and $c_{1/2}$ are unknown, while the constant is predicted analytically and should not depend on the universality class. The coefficients $c_{3/2}$ and $c_{1/2}$ are universal, but differ between different universality classes.

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP³.

The dimensions D_q for $q \leq 6$ have been computed directly by using various methods, in particular Monte Carlo simulations of lattice models, field theoretic methods, and the conformal bootstrap. Irrespective of the method, the error rapidly increases with increasing q . For a summary and a discussion of these results see Sec. V.

In Ref. [14] an approach is suggested that allows to reach larger values of q by using Monte Carlo simulations of lattice models. The authors simulate the standard XY model on the simple cubic lattice by using the worm algorithm [15,16]. They study the two-point function at criticality:

$$C_q(r) = \langle \exp(-iq(\theta_0 - \theta_r)) \rangle \simeq |r|^{-2D_q}, \quad (2)$$

where r is a site on the finite lattice with the linear extension L . In order to cancel finite size effects, they consider $r = (L/2, 0, 0)$ throughout. In a direct approach the error of D_q , for a given numerical effort, would increase rapidly with q . The way out is to compute ratios

$$R_q(r) = \frac{C_q(r)}{C_{q-1}(r)} \simeq |r|^{-2\Delta_q}, \quad (3)$$

where $\Delta_q = D_q - D_{q-1}$. A discussion of the implementation is given below in Sec. III. The authors report results up to $q = 12$, see Table II of Ref. [14]. More recently the idea was picked up in Ref. [17] who added a new algorithmic idea and studied three-point functions in addition. The authors of [17] reach $q = 19$, albeit with a rather big error of Δ_q . The numerical results are in good agreement with Eq. (1) down to astonishingly small values of q .

Here we build on Ref. [14] and use the algorithmic idea of Ref. [17]. However, instead of the standard XY model on the simple cubic lattice, we simulate the improved model studied in Ref. [18]. Improved means that the amplitude of the leading correction to scaling vanishes. Major progress is achieved by using nontrivial weights in the worm update [19]. This allows us to reach $q = 64$ at high numerical precision. Our results show that both Refs. [14,17] have underestimated systematic errors due to corrections to scaling. Our errors on Δ_q are in the fourth digit. The results are in excellent agreement with Eq. (1) down to $q \approx 4$. The deviation is small even for $q = 1$. This certainly calls for a better theoretical understanding.

The outline of the paper is the following: First, in Sec. II we define the 3N XY model that we simulate and the quantities that we measure. Then in Sec. III we explain how the 3N XY model is simulated by using the worm algorithm and how the ratio of two-point functions is implemented. In Sec. IV we discuss the simulations and the analysis of the resulting data. Section IV starts with its own outline. Finally, we summarize our results and compare with the literature. In the Appendix we provide some additional discussion on the worm algorithm.

II. THE MODEL AND THE TWO-POINT FUNCTION

We consider the XY model on the simple cubic lattice with nearest neighbor and next-to-next-to-nearest neighbor (3N) couplings. Below we shall denote the model by 3N XY model. The reduced Hamiltonian is given by

$$\mathcal{H}[\{\vec{s}\}] = -K_1 \sum_{\langle xy \rangle} \vec{s}_x \cdot \vec{s}_y - K_3 \sum_{[xy]} \vec{s}_x \cdot \vec{s}_y, \quad (4)$$

where the spin or field variable $\vec{s}_x$ is a unit vector with two real components. It can be written as $\vec{s}_x = [\cos(\theta_x), \sin(\theta_x)]$, where $\theta_x \in [0, 2\pi)$. $x = (x_0, x_1, x_2)$ denotes a site on the simple cubic lattice, where $x_i \in \{0, 1, \dots, L_i - 1\}$. Furthermore, $\langle xy \rangle$ denotes a pair of nearest and $[xy]$ a pair of next-to-next-to-nearest, or third nearest (3N) neighbors on the lattice. In our simulations, the linear lattice size $L = L_0 = L_1 = L_2$ is equal in all three directions and periodic boundary conditions are employed. The partition function is given by

$$Z = \int \mathcal{D}[\{\vec{s}\}] \exp(-\mathcal{H}[\{\vec{s}\}]). \quad (5)$$

In Ref. [18] we have set up the FSS study such that we pass the critical line for a given value of K_3 by varying K_1 . Hence we get a critical coupling $K_{1,c}$ that depends on K_3 . The standard XY model is obtained for $K_3 = 0$.

We find that leading corrections to scaling at criticality are eliminated at $K_3^* = 0.04149(14)$. For $K_3 = 0.0415$ we

get $K_{1,c} = 0.37069947(2)$. Below, we shall simulate at $(K_1, K_3) = (0.37069947, 0.0415)$. Note that in Ref. [18] we have actually simulated the q -state clock model. For the values of q that we have simulated, the difference in $K_{1,c}$ compared with the limit $q \rightarrow \infty$ is negligible.

The choice of the improved model is not unique. In the literature, for example, the two-component ϕ^4 model and the dynamically diluted XY model are studied. In Refs. [18,20–23] the values of the parameters, where leading corrections to scaling vanish are accurately determined.

Here we have selected the 3N XY model, since it is straight forward to use the worm algorithm as it has been implemented for the standard XY model in the literature. Furthermore in Ref. [18] we have determined $K_{1,c}$ and K_3^* very accurately and corrections due to the violation of the rotational invariance are reduced compared with the standard XY model.

There are different ways to extract the dimensions of fields in Monte Carlo simulations of lattice models. Here we follow the straight forward approach that has been used in Refs. [14,17]. At criticality the two-point function, Eq. (2), behaves as $C_q(r) \simeq |r|^{-2D_q}$, where D_q is the dimension of the field. For a finite lattice this behavior is observed only for $|r| \ll L$, where L is the linear lattice size. In practice, this requirement impedes precision estimates of D_q . Therefore, following Ref. [14], we chose $|r| = cL$ to cancel finite size effects. This means, for a given linear lattice size L , only one distance is considered. In the case of Ref. [14] $r = (L/2, 0, 0)$. Here we decided *ad hoc* to use the maximal distance on the torus

$$r = (L/2, L/2, L/2). \quad (6)$$

We did not carefully compare these two choices. Likely, the difference in statistical errors and systematic errors due to corrections to scaling is rather small. The two-point function can be determined by using the worm algorithm as we outline below.

III. THE WORM ALGORITHM FOR THE XY MODEL WITH 3N COUPLINGS

Here we follow previous work on the standard XY model, for example Refs. [14,24,25]. Taking care of the additional 3N coupling is straight forward. In order to discuss the worm algorithm, the partition function has to be rewritten. A new field $\{k\}$ that lives on the edges of the lattice is introduced, which allows to integrate out the original field $\{\vec{s}\}$ that lives on the sites. In the case of our model we have to add variables for the 3N neighbor pairs. First we express the reduced Hamiltonian in terms of angles

$$H(\{\theta\}) = -K_1 \sum_{\langle xy \rangle} \cos(\theta_x - \theta_y) - K_3 \sum_{[xy]} \cos(\theta_x - \theta_y), \quad (7)$$

where we change variables as $\vec{s}_x = [\cos(\theta_x), \sin(\theta_x)]$ with $\theta_x \in [0, 2\pi)$. It holds

$$\exp(K \cos(\theta)) = \sum_{k=-\infty}^{\infty} I_k(K) \exp(ik\theta), \quad (8)$$

where $I_k(K)$ is the modified Bessel function of the first kind. Hence the partition function becomes

$$\begin{aligned} Z &= \int D[\theta] \left[\prod_{\langle xy \rangle} \exp(K_1 \cos(\theta_x - \theta_y)) \right] \left[\prod_{[xy]} \exp(K_3 \cos(\theta_x - \theta_y)) \right] \\ &= \int D[\theta] \sum_{\{k\}} \left[\prod_{\langle xy \rangle} I_{k_{\langle xy \rangle}}(K_1) \exp(ik_{\langle xy \rangle}[\theta_x - \theta_y]) \right] \left[\prod_{[xy]} I_{k_{[xy]}}(K_3) \exp(ik_{[xy]}[\theta_x - \theta_y]) \right] \\ &= \sum_{\{k\}} \left[\prod_{\langle xy \rangle} I_{k_{\langle xy \rangle}}(K_1) \int D[\theta] \exp(ik_{\langle xy \rangle}[\theta_x - \theta_y]) \right] \left[\prod_{[xy]} I_{k_{[xy]}}(K_3) \int D[\theta] \exp(ik_{[xy]}[\theta_x - \theta_y]) \right], \end{aligned} \quad (9)$$

where $\{k\}$ is a configuration of the k variables. We need a convention, for the orientation of the edges $\langle xy \rangle$ and $[xy]$. Depending on the relative orientation, going from x to y , -1 or 1 is multiplied on $k_{\langle xy \rangle}$ or $k_{[xy]}$. Hence $k_{\langle xy \rangle} = -k_{\langle yx \rangle}$ and $k_{[xy]} = -k_{[yx]}$. In the case of $\langle xy \rangle$ we assign the positive orientation to the upward link. In the case of $[xy]$, the positive orientation goes to x_0 up and the negative one to x_0 down. Of course, equivalently we could take x_1 or x_2 to this end. Performing the integral over θ , we arrive at the constraint

$$\sum_{y.nn.x} k_{\langle xy \rangle} + \sum_{y.3n.x} k_{[xy]} = 0 \quad (10)$$

for all x . Here $y.nn.x$ means that y is a nearest neighbor of x and $y.3n.x$ that y is a third nearest neighbor of x . The constraint is solved by packing closed oriented loops on the lattice. For our model, the loop might run through the edges $\langle xy \rangle$ as for the standard XY model with $K_3 = 0$, but now in addition through the diagonal edges $[xy]$. A loop that runs through $\langle xy \rangle$ or $[xy]$ from x to y contributes 1 to $k_{\langle xy \rangle}$ or $k_{[xy]}$ and hence -1 to $k_{\langle yx \rangle}$ or $k_{[yx]}$.

In the worm algorithm, the ensemble is enlarged. In addition to the partition function as given above, there are unnormalized two-point functions:

$$Z_{u,v} = \int D[\{\vec{s}\}] \exp(-\mathcal{H}[\{\vec{s}\}]) \vec{s}_u \cdot \vec{s}_v, \quad (11)$$

where $\vec{s}_u \cdot \vec{s}_v = \text{Re} \exp(-i[\theta_u - \theta_v])$. In the following, u is the tail and v the head of the worm. We note that $Z_{u,u} = Z$, since $\vec{s}_u \cdot \vec{s}_u = 1$. We can proceed as for the partition function above. The only difference is that the constraint, Eq. (10), is modified for the sites u and v , with $u \neq v$

$$\sum_{y.nn.u} k_{\langle uy \rangle} + \sum_{y.3n.u} k_{[uy]} = 1 \quad (12)$$

and

$$\sum_{y.nn.v} k_{\langle vy \rangle} + \sum_{y.3n.v} k_{[vy]} = -1. \quad (13)$$

In addition to the closed loops, there is now an oriented line, the worm, with the two ends u and v .

The enlarged ensemble is given by

$$Z_{\Sigma} = \sum_{u,v} w(u,v) Z_{u,v}, \quad (14)$$

where we have introduced the weights $w(u,v) > 0$ following Ref. [19]. Below we shall demonstrate that a suitable $w(u,v)$ allows to obtain precision results even for large values of q . Let us write Z_{Σ} as a sum over configurations

$$Z_{\Sigma} = \sum_{u,v} w(u,v) \sum_{\{k\}'} \left[\prod_{\langle xy \rangle} I_{k_{\langle xy \rangle}}(K_1) \right] \left[\prod_{[xy]} I_{k_{[xy]}}(K_3) \right]. \quad (15)$$

A configuration consists of the location of u and v and a choice $\{k\}'$ of $\{k\}$ that satisfies the constraints are Eqs. (10), (12), (13). Let us recall the two steps, following Ref. [19], of the worm update:

- (i) Pick one of the neighbors of v with equal probability for each of the neighbors as proposal v' for the new head site. For our model, there are 14 possible choices. Along with that $k'_{<v,v'>} = k_{<v,v'>} + 1$ or $k'_{[v,v']} = k_{[v,v']} + 1$ is taken.

The site v' along with the k' is accepted with the probability

$$P_{\text{acc}} = \min[1, w(u,v') I_{k'}(K_i) / (w(u,v) I_k(K_i))], \quad (16)$$

where $i = 1$ or 3 for v' nearest or next-to-next-to-nearest neighbor.

- (ii) If $u = v$, chose a new tail that is equal to the head according to some rule that makes this step stable, where stable means that it conserves the correct distribution of the $Z_{u,u}$.

The first step fulfils detailed balance. In the second step we jump from $u = v$ to a new starting point of the worm. The simplest choice is to chose the new site with an equal probability for each site of the lattice. Below, our measurement requires that $u = (0, 0, 0)$. Hence we should insert tail and head at $u = (0, 0, 0)$ much more frequently than at other sites of the lattice. This can be formally encoded by using the weight $w(u, u)$. It is much larger for $u = (0, 0, 0)$ than for other sites.

The most simple quantity to determine in the worm algorithm is the two-point function

$$\langle \vec{s}_x \cdot \vec{s}_y \rangle = \frac{Z_{x,y}}{Z_{x,x}} = \frac{w(x, x) \langle \delta_{(x,y),(u,v)} \rangle}{w(x, y) \langle \delta_{(x,x),(u,v)} \rangle}. \quad (17)$$

Next let us discuss the algorithmic idea discussed in the Appendix A of Ref. [17]. The elementary step of the worm update is equivalently reorganized. The authors denote their version of the worm algorithm as continuous time (CT) update.

If we are at $v \neq u$, eventually we are moving to a new head v' . For each of the nearest and $3N$ nearest neighbors v' there is a probability

$$p(v', v) = \frac{1}{14} \min[1, w(v') I_{k'}(K_i) / (w(v) I_k(K_i))] \quad (18)$$

to be the successor of v . In our case $1/14$ is the probability to be selected as proposal and then follows the acceptance probability.

The probability to get a new head of the worm in one step is

$$P_{\text{move}} = \sum_{v'.nn.v} p(v', v) + \sum_{v'.3n.v} p(v', v) \leq 1. \quad (19)$$

If no new head v' is accepted, one tries again and so on. On average one stays at the site v for the time

$$t_v = \sum_{i=1}^{\infty} i P_{\text{move}} (1 - P_{\text{move}})^{i-1} = 1/P_{\text{move}}. \quad (20)$$

Note that

$$\sum_{i=1}^{\infty} P_{\text{move}} (1 - P_{\text{move}})^{i-1} = 1 \quad (21)$$

meaning that we will move on at some time for sure.

In the algorithm, we go ahead straight away, taking into account t_v , when computing observables. The new head is chosen among the 14 candidates in the following way: First we label the candidates by $\alpha = 1, 2, \dots, 14$ and define $\tilde{p}_0 = 0$ and

$$\tilde{p}_\alpha = \sum_{\gamma=1}^{\alpha} p(v_\gamma, v) / P_{\text{move}} \quad (22)$$

Note that $\tilde{p}_{14} = 1$. Now α is selected in the following way: We draw a random number r that is uniformly distributed in $[0, 1)$. Then α is given by

$$\tilde{p}_{\alpha-1} \leq r < \tilde{p}_\alpha. \quad (23)$$

We have to evaluate $p(v_\gamma, v)$ for all 14 neighbors. However, only one pseudorandom number is needed. On the other hand, for the original version of the update, we need on average t_v times two random numbers along with the evaluation of $p(v', v)$.

How these two choices compare, depends on the CPU times that are needed for the generation of a random number, the evaluation of $p(v', v)$ and the values of $p(v', v)$. In our case we find a speed up by factor slightly smaller than two by using the version of Ref. [17].

The question is how to deal with the insertions of the tail site. Here in our simulations, we left the update step in its original form, when the worm is newly inserted, and only used the CT version for $v \neq u$. This way, for updating the $\{k\}$, we can chose a new site for the tail. The authors of Ref. [17] just keep the tail at $u = 0$ and let the head run forever. This is a viable choice.

A. Ratios $R_q(r) = C_q(r)/C_{q-1}(r)$

In principle the two-point functions $C_q(r)$ can be computed directly using the worm algorithm, by using heads and tails with the corresponding charges. However, the authors of [14] observed that the estimator of $C_q(r)$ becomes increasingly noisy with increasing q . The way out proposed by them is a divide and conquer strategy: Instead of computing $C_q(r)$ directly, ratios $R_q(r) = C_q(r)/C_{q-1}(r)$ are computed. To this end one simulates

$$Z_\Sigma = \sum_v Z_0 \exp[-i(q-1)(\theta_0 - \theta_r)] \exp[-i(\theta_0 - \theta_v)]. \quad (24)$$

using the worm algorithm.

The simulation is started by preparing a $\{k\}$ configuration with charge $-q + 1$ at $x = 0$ and $q - 1$ at r . In the case of $r = (L/2, 0, 0)$ we could for example set

$$k_{\langle xy \rangle} = q - 1 \quad (25)$$

for all $x = (x_0, 0, 0)$, $y = (x_0 + 1, 0, 0)$, where $x_0 \in \{0, 1, 2, \dots, L/2 - 1\}$. For all other $\langle xy \rangle$ we set $k_{\langle xy \rangle} = 0$. Obviously other more complicated paths could be taken or the charge could be distributed over many different paths between 0 and r . In the case of the 3N model we could also use diagonal edges to this end. Here, for $r = (L/2, L/2, L/2)$ we used $\langle (x_0, x_0, x_0), (x_0 + 1, x_0, x_0) \rangle$, $\langle (x_0 + 1, x_0, x_0), (x_0 + 1, x_0 + 1, x_0) \rangle$, and $\langle (x_0 + 1, x_0 + 1, x_0), (x_0 + 1, x_0 + 1, x_0 + 1) \rangle$ with $x_0 \in \{0, 1, 2, \dots, L/2 - 1\}$.

The updates of the system are done in the same way as for $q = 0$. However, for measuring $R_q(r)$ we have to set the tail of the worm to the site $(0, 0, 0)$. The ratio we are looking for is obtained by

$$R_q(r) = \frac{w(0, 0)}{w(0, r)} \langle \delta_{v,r} \rangle_{q-1}. \quad (26)$$

This means, we just have to count, how often the head of the worm visits the site r . In the case of the continuous time version we have to add t_r , when the site r is visited.

As nontrivial choice of the weight we used

$$w(0, x) \approx R_q^{-1}(x) = \langle \vec{s}_0 \cdot \vec{s}_x \rangle_{q-1}^{-1}. \quad (27)$$

The idea is that the head of the worm visits the sites of the lattice with equal probability for each of the sites. We did not use an analytic expression for the two-point function, but used preliminary numerical estimates instead. For small values of q , we run the worm algorithm using $w(0, x) = 1$ to this end. For large q we iterate this procedure. As starting point we use the results for smaller values of q that were obtained before. For the estimate of $R_q(x)$ that is used in the production runs we performed simulations that take order one day on a single CPU core for $L = 128$. This are about 2000000 single worm updates. We did not systematically study the question, how many updates are optimal here. We just performed as many updates as can be done in a reasonable time. Test runs for smaller lattice sizes, varying the number of measurements, suggest that we are close to optimal this way.

Sampling and storing the result, we make use of the symmetries of the lattice, taking into account the insertions of the charges at $x = 0$ and $x = r$. In the program we exploit

$$R_q(x_0, x_1, x_2) = R_q(\pm x_0, \pm x_1, \pm x_2), \quad (28)$$

where the arguments are understood modulo L and permutations of the components x_0, x_1 and x_2 .

In our production runs, we first performed a certain number n_{up} of worm updates for using $w(u, x) = \text{const}$ with $u \neq (0, 0, 0)$ for each measurement worm with $u = (0, 0, 0)$ and the nontrivial weight $w(0, x)$. Later, in particular for large q , we skipped the worm updates with

$u \neq (0, 0, 0)$, since this way, we still can generate all possible loop configurations. For lack of time, we did not carefully benchmark the performance of these different options.

B. Gain by using nontrivial weights

Here we jump ahead of the discussion of our simulations and already report the gain that is achieved by using $w(0, x) \approx R_q^{-1}(x)$. We define the efficiency of the update as

$$E_{\text{algo}}(q, L) = 1/([\text{CPU time needed}] \times [\text{stat error}]^2). \quad (29)$$

We have benchmarked both versions of the update, $w(0, x) = 1$ and the nontrivial $w(0, x) = R_q^{-1}(x)$ on the same PC at $(K_1, K_3) = (0.37069947, 0.0415)$. To compare the two choices, we compute the ratios $r_{\text{eff}} = E_{w(0,x)=\text{nontrivial}}(q, L)/E_{w(0,x)=1}(q, L)$, which should not depend much on the precise version of the CPU. To this end we take the programs used in the simulation, which are reasonably well tuned. We have computed r_{eff} for a small selection of q and L : For $q = 2$ we get $r_{\text{eff}} = 1.64$ and 2.3 for $L = 32$ and 96 , respectively. For $q = 4$ we get $r_{\text{eff}} = 6.1, 8.0$, and 10.2 for $L = 32, 64$, and 96 , respectively. In the case of $q = 8$ we performed simulations using $w(x, 0) = 1$ only for small lattice sizes. For example, we get $r_{\text{eff}} = 11.7$ and 16.8 for $L = 16$ and 32 , respectively.

For given q , the gain by using a nontrivial weight $w(0, x)$ grows slowly with increasing lattice size L . For given L there is a rapid increase with q . We abstain from computing the gain for larger values of q . But there can be little doubt that it further increases.

IV. NUMERICAL RESULTS

In Sec. IV A we discuss how we estimate the derivative of $R_q(L, K_1)$ with respect to K_1 . This derivative is needed to estimate the error in our estimates of Δ_q due to the error in $K_{1,c}$. Next in Sec. IV B we discuss leading corrections, and how the error due to residual leading corrections to scaling at $K_3 = 0.0415$ is estimated. In Sec. IV C we summarize the *Ansätze* that are used to fit the data for $R_q(L)$ obtained in our simulations. In Sec. IV D we discuss our simulations of the standard XY model. These are performed mainly to estimate the effect of leading corrections on $R_q(L)$. The results of this Sec. are needed to estimate the error due to residual leading corrections at $K_3 = 0.0415$. Then, in Sec. IV E we discuss the simulations of the improved 3N XY model and the analysis of the $R_q(L)$ data which results in accurate estimates of Δ_q . Finally in Sec. IV F, we fit our estimates of Δ_q to *Ansätze* based on Eq. (1) to obtain estimates of the coefficients $c_{3/2}$ and $c_{1/2}$. Next, for $q \leq 8$, we compute D_q by summing up our estimates of Δ_q . Using these estimates, we compute

$\Delta_q - c_{3/2}q^{3/2} - c_{1/2}q^{1/2}$, which is compared with the constant part of Eq. (1).

A. Derivative with respect to K_1

The derivative of $R_q(L, K_1)$ with respect to K_1 is needed to estimate the error of Δ_q due to the uncertainty of $K_{1,c}$. To this end, no high precision estimate is needed.

The derivative is estimated by using the finite difference

$$S_{R_q}(L, K_{1,c}) = \left. \frac{\partial R_q(L, K_1)}{\partial K_1} \right|_{K_1=0.37069947} \approx \frac{R_q(L, K_{1,a}) - R_q(L, K_{1,b})}{K_{1,a} - K_{1,b}}, \quad (30)$$

where $K_{1,a} > K_{1,c}$ and $K_{1,b} < K_{1,c}$ are close to the critical coupling. The derivative behaves as

$$\frac{S_{R,q}(L, K_{1,c})}{R_q(L, K_{1,c})} = a_q L^{y_t}, \quad (31)$$

Having determined an estimate of a_q , we obtain rough estimates of $R_q(L, K_1)$ in the neighborhood of $R_q(L, K_{1,c})$ by using

$$R_q(L, K_1 + \Delta K_1) = R_q(L, K_1)(1 + \Delta K_1 a_q L^{y_t}). \quad (32)$$

Throughout, for all values of q , we simulated at $K_1 = 0.3700$ and 0.3714 for the linear lattice size $L = 16$. In the case of $q = 4$, as check, we varied the lattice size and the values of K_1 . We get consistent results.

The error induced on Δ_q is estimated by redoing fits using the data for R_q multiplied by $1 + \Delta K_1 a_q L^{y_t}$, where ΔK_1 is the error of $K_{1,c}$ now.

B. Residual leading corrections to scaling

The value of K_3^* is only approximately known. Therefore there might be a small amplitude of the leading correction at $K_3 = 0.0415$.

In Ref. [18] we have determined $K_3^* = 0.04149(14)$ by analyzing dimensionless quantities. Let us briefly recall the logic behind using improved models in a simplified setting: Let us take the Binder cumulant and its slope as example. We assume for simplicity that the critical temperature is exactly known. Then in general

$$U_4(L, K_{1,c}, K_3) = U_4^* + b(K_3)L^{-\omega} + c[b(K_3)L^{-\omega}]^2 + \dots + d(K_3)L^{-\epsilon} + \dots \quad (33)$$

To further simplify the argument, we focus on the leading correction. In the fit we are using the *Ansatz*

$$U_4(L, K_{1,c}, K_3) = U_4^* + b(K_3)L^{-\omega}. \quad (34)$$

Furthermore, we assume that we know ω from other sources. Hence, for a given value of K_3 , the parameters of the fit are U_4^* and $b(K_3)$. Say in our study we are aiming at the RG exponent y_t . In finite size scaling, we extract it from the slope of dimensionless quantities. Here we take the Binder cumulant as example

$$S_{U_4}(L, K_{1,c}, K_3) = \left. \frac{\partial U_4(L, K_{1,c}, K_3)}{\partial K_1} \right|_{K_1=K_{1,c}} = a(K_3)L^{y_t}(1 + pb(K_3)L^{-\omega} + \dots). \quad (35)$$

Say we simulate only the XY model, $K_3 = 0$. Then we have three free parameters in the fit: $a(0)$, y_t , and $pb(0)$. One has to keep in mind that the more free parameters there are in the *Ansatz*, the larger is the statistical error of the parameters in the fit. Hence, by fitting U_4 we get $b(K_3)$ more accurately than $pb(K_3)$ by fitting S_{U_4} .

Simulating at $K_3 \approx K_3^*$, we know that $pb(K_3^*)$ is small. However, simply ignoring the correction term in the *Ansatz* results in a systematic error in our estimate of y_t , which is likely small. However, we would like to know how small. Here information from the simulation of a model with a reasonably large leading correction amplitude, say the XY model, helps. Fitting the estimates of U_4 and its slope for the XY model, we get estimates of $b(K_3)$ and $pb(K_3)$ and hence $p = [pb(K_3)]/b(K_3)$.

Analyzing the Binder cumulant U_4 for the XY and the 3N XY model at $K_3 = 0.0415$ and 0.046 , using the data of Refs. [18,23] we translate the error in $K_3^* = 0.04149(14)$ to the bound $|b(0.0415)/b(0)| \lesssim 1/250$. Hence the amplitude of the leading correction in S_{U_4} at $K_3 = 0.0415$ at criticality is bounded by

$$|pb(0.0415)| \lesssim |pb(0)|/250. \quad (36)$$

Now we estimate the possible error of y_t due to leading corrections as follows. First we fit our data for $K_3 = 0.0415$ with the *Ansätze*

$$S(L) = aL^{y_t} \quad (37)$$

and for example

$$S(L) = aL^{y_t}(1 + cL^{-\epsilon}), \quad (38)$$

where $\epsilon \approx 2$ corresponding to subleading corrections. Here we take our data for $S(L)$ obtained in the simulations. Next we repeat these fits, replacing the data for $S(L)$ by the data multiplied by $1 + (|pb(0)|/250)L^{-\omega}$. The difference between these two sets of fits provides us with an estimate of the error due to residual leading corrections at $K_3 = 0.0415$. Below, analyzing $R_q(L)$, we proceed in an analogous way.

C. Fit *Ansätze*

In the case of the 3N XY model, we performed fits by using the *Ansätze*

$$R_q(L) = a(q)L^{-2\Delta_q}, \quad (39)$$

$$R_q(L) = a(q)L^{-2\Delta_q}[1 + c(q)L^{-2}] \quad (40)$$

and

$$R_q(L) = a(q)L^{-2\Delta_q}[1 + c(q)L^{-2} + d(q)L^{-4}]. \quad (41)$$

The correction $\propto L^{-2}$ is due to second derivatives of the field with respect to the location. See, for example, Appendix C of Ref. [26]. There should be corrections due to the violation of the rotational invariance by the lattice. The corresponding correction exponent is $\omega_{NR} = 2.02548(41)$ [27,28]. In the 3N XY model at $K_3 = 0.0415$ the amplitude of this correction is suppressed by approximately a factor of 9 compared with the standard XY model [18]. In the fits we expect that this correction is effectively taken into account by the term $\propto L^{-2}$. The correction $\propto L^{-4}$ is taken *ad hoc*. It is mainly justified by the fact that it nicely fits the data. Comparing the results obtained by using the different *Ansätze* provides us with an estimate of the systematic error caused by higher order corrections to scaling. It turns out that the amplitudes of corrections to scaling increase with increasing q . Therefore the *Ansatz* (41) is useful in particular for larger values of q .

In the case of the standard XY model we used as above Eq. (39) and in addition

$$R_q(L) = a(q)L^{-2\Delta_q}[1 + b(q)L^{-\omega}] \quad (42)$$

and

$$R_q(L) = a(q)L^{-2\Delta_q}[1 + b(q)L^{-\omega} + c(q)L^{-2}] \quad (43)$$

to take leading corrections to scaling into account.

D. The standard XY model

Since the leading correction is not exactly eliminated in the 3N XY model, we need simulations of the standard XY to estimate the residual amplitude of the leading correction in the 3N XY model. We simulated the standard XY model at $K = K_1 = 0.45416475$. Note that $K_{1,c} = 1/2.2018441(5) \approx 0.45416476(10)$ given in Ref. [25] and $K_{1,c} = 0.45416474(10)[7]$ [23]. Here we simulated for $q = 2, 3, 4, 6, 8, 16$ and 32 .

As an example let us discuss the analysis of our data for $q = 4$ in detail. We simulated the linear lattice sizes $L = 6, 8, \dots, 20, 24, \dots, 32, 48, 64$, and 96 . We had started with simulations with $w(0, x) = 1$ and had reached a statistics that we regarded as final. Later, having realized the gain

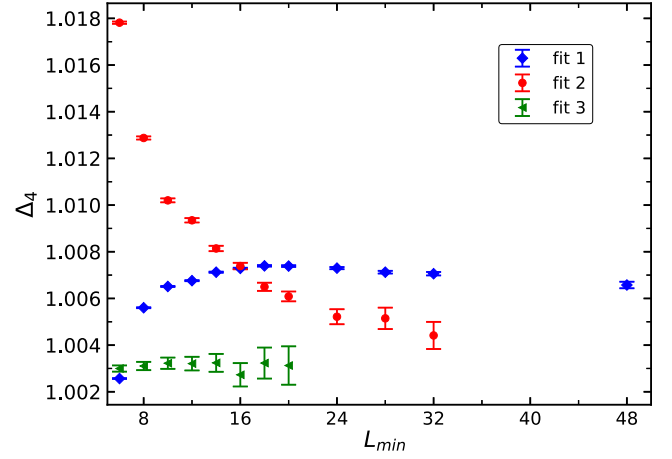


FIG. 1. Data for $R_4(L)$ for the standard XY model at $K = 0.45416475$ are analyzed. Estimates of Δ_4 are plotted versus the minimal linear lattice $L_{\min}$ that is taken into account in the fit. The estimates are obtained by using the *Ansätze* (39), (42), and (43). In the caption we denote the fits by fit 1, fit 2, and fit 3, respectively. Details are discussed in the text.

that could be achieved by the proper choice of $w(0, x)$, we have redone the simulations, now achieving a better accuracy. Using $w(0, x) \approx R_q^{-1}(x)$, we performed for example 10^{10} measurements for $L = 16$. This number decreases to 7.3×10^8 for $L = 96$. The simulations took about the equivalent of 1.1 years of CPU time on a single core of an Intel(R) Xeon(R) CPU E3-1225 v3 running at 3.20 GHz for $w(0, x) \approx R_q^{-1}(x)$ in total. We checked that the results obtained in these two sets of simulations are consistent. We merged the data before fitting. In Fig. 1 we plot the estimates of Δ_q obtained by using the *Ansätze* (39), (42), and (43). Note that we get fits with $\chi^2/\text{DOF} \approx 1$ for the *Ansatz* (43) starting from $L_{\min} = 6$. In the case of the *Ansatz* (42), we get $\chi^2/\text{DOF} = 1.73$ for $L_{\min} = 20$ but it does not further drop. For the pure power law, Eq. (39), we get $\chi^2/\text{DOF} = 9.87$ for $L_{\min} = 18$ but χ^2/DOF does not further decrease toward clearly smaller values, when increasing $L_{\min}$. In our fits all linear lattice sizes $L \geq L_{\min}$ are taken into account. Fitting with the *Ansatz* (43) gives a stable result starting from the smallest lattice sizes considered here. Here we are not quoting a final estimate. To this end, we should take into account that there are corrections proportional to $L^{-2\omega}$, which are not included into the *Ansatz*. Also the error due to the uncertainty of the correction exponent ω should be taken into account in the estimate of the systematic error. But let us keep in mind that for example for $L_{\min} = 12$, using the *Ansatz* (43), we get $\Delta_4 = 1.0032(3)$ just to see how it compares with the estimate obtained from the improved model. In the following, we need an estimate of $b(4)$, Eqs. (42) and (43), to estimate the error due to residual leading corrections in the case of the 3N XY model. Based on the *Ansatz* (43) we get $b(4) = 0.27(7)$. For the other values of q we proceeded in a

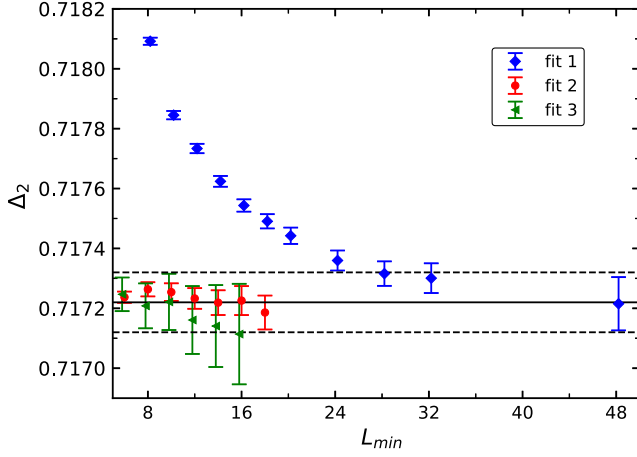


FIG. 2. Data for the 3N XY model at $K_1 = 0.37069947$ and $K_3 = 0.0415$ are analyzed. Estimates of Δ_2 obtained by fitting $R_2(L)$ using the *Ansätze* (39), (40), and (41) are plotted versus the minimal linear lattice size $L_{\min}$ that is taken into account. In the caption we denote the fits by fit 1, fit 2, and fit 3, respectively. The solid line indicates our final estimate, while the dashed lines give the error. The values of $L_{\min}$ are slightly shifted to avoid overlap of the symbols. Details are discussed in the text.

similar fashion. We find $b = 0.00(6)$, $0.18(10)$, $0.55(9)$, $0.74(6)$, $1.56(6)$, $3.3(1)$, for $q = 2, 3, 6, 8, 16$, and 32 , respectively. We note that the amplitude of corrections vanishes within error bars for $q = 2$. Roughly it increases linearly with increasing q . Below in Sec. IV E we shall use estimates based on a linear fit $b(q) = -0.165 + 0.1087q$ for the values of q that we did not simulate for the standard XY model.

E. 3N XY model

As discussed in Sec. II, we simulated the 3N XY model at $(K_1, K_3) = (0.37069947, 0.0415)$. For the 3N XY model, we performed simulations for $q = 2, 3, \dots, 8, 16, 24, 32, 48$ and 64 . For $q = 2$ and 3 , we simulated the linear lattice sizes $L = 6, 8, 10, 12, 14, 16, 18, 20, 24, 28$,

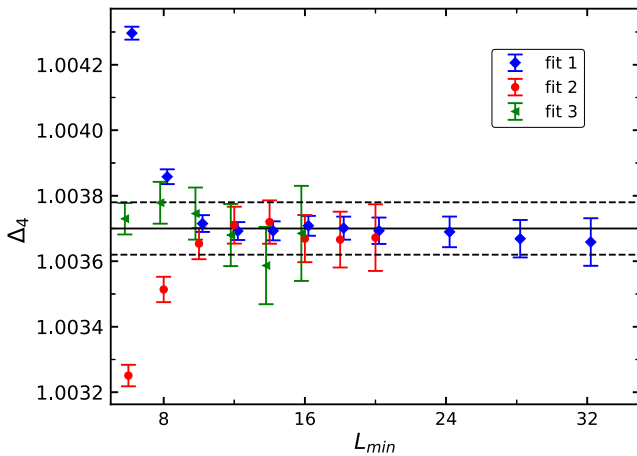


FIG. 3. Same as Fig. 2 but for $q = 4$ instead of $q = 2$.

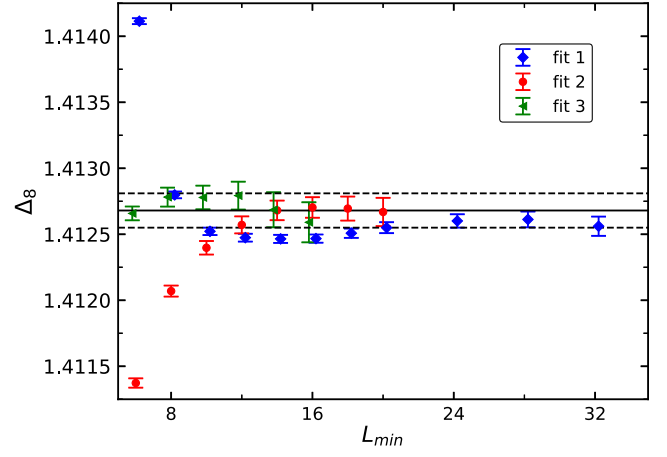


FIG. 4. Same as Fig. 2 but for $q = 8$ instead of $q = 2$.

$32, 48, 64, 96$, and 128 . For $q = 4, 5, \dots, 8$ we omit $L = 128$. For $q = 16, 24$ and 32 we start from $L = 12$ and simulate up to $L = 128$, while for $q = 48$ and 64 we start from $L = 16$ and simulate up to $L = 128$. The lattice sizes in between are chosen as above for $q = 2$ and 3 . For small q , we have generated data using $w(0, x) = 1$ before switching to nontrivial $w(0, x)$. We checked that the results obtained in these two sets of simulations are consistent. We merged the data before fitting.

For $q = 2$ and the nontrivial $w(0, x)$, we performed about 4×10^{10} measurements for $L = 16$. This number decreases to 1.7×10^9 for $L = 128$. For example for $L = 128$ we get $R_2(128) = 0.00048747(10)$. For larger q the statistics is lower. The simulations for $q = 2$ and 3 took the equivalent of about 3.3 and 2.5 years of CPU time on a single core of an Intel(R) Xeon(R) CPU E3-1225 v3 running at 3.20 GHz for $w(0, x) \approx R_q^{-1}(x)$ in total, respectively. For $q \geq 4$, we used slightly more than one year of CPU time for each value of q .

In Figs. 2–7 we plot our results for Δ_q obtained by fitting $R_q(L)$ with the *Ansätze* (39), (40), and (41) for $q = 2, 4, 8$,

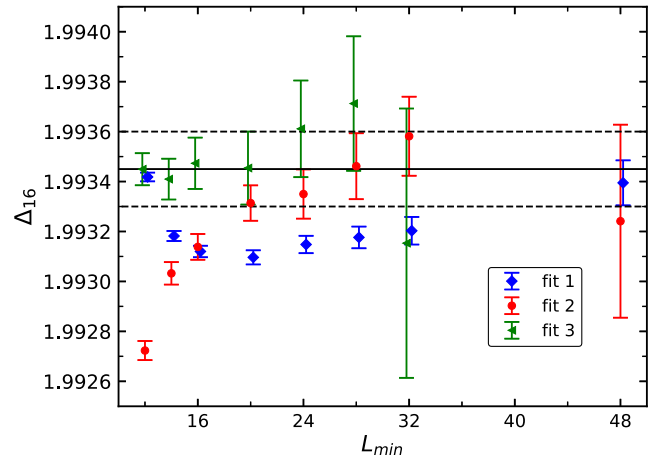
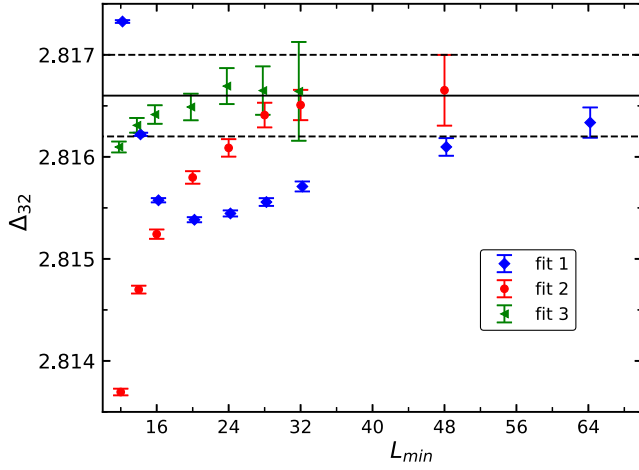


FIG. 5. Same as Fig. 2 but for $q = 16$ instead of $q = 2$.

FIG. 6. Same as Fig. 2 but for $q = 32$ instead of $q = 2$.

16, 32, and 64, respectively, versus $L_{\min}$. Note that in these fits all linear lattice sizes with $L \geq L_{\min}$ are taken into account. Typically in such fits χ^2/DOF has a large value for small $L_{\min}$ and decreases rapidly with increasing $L_{\min}$ reaching $\chi^2/\text{DOF} \approx 1$. In Table I we give the $L_{\min}$, where this point is reached depending on q and the *Ansatz* that is used. Let us denote this minimal lattice size by $L_{\min,\min}$. Let us denote fits with $L_{\min} \geq L_{\min,\min}$ as acceptable fits. Note that this says little on systematic errors on the parameters of the fit due to corrections that are not taken into account in the *Ansatz*. In order to get some handle on systematic errors, we require that our final estimate is consistent with fits using different *Ansätze*. The final estimate is chosen such that it covers for each of the *Ansätze* (39), (40), and (41) at least the estimate obtained by one acceptable fit. By $q \pm \Delta q$ covers $p \pm \Delta p$ we mean that $q + \Delta q \geq p + \Delta p$ and $q - \Delta q \leq p - \Delta p$ hold, where Δp is the error of the parameter p . In the figures our final result is shown as solid line and the error is indicated by dashed lines. Our final estimates are summarized in Table II.

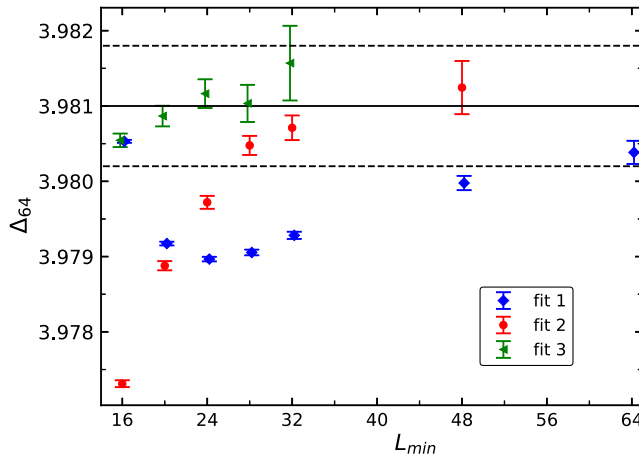
FIG. 7. Same as Fig. 2 but for $q = 64$ instead of $q = 2$.

TABLE I. We give $L_{\min,\min}$ as a function of q and the *Ansatz*. For the definition of $L_{\min,\min}$ see the text. The values of q correspond to the Figs. 2–7. Note that for $q = 2, 4$, and 16 the lattice sizes $L = 6, 6$ and 12 are the smallest that we have simulated. For a discussion see the text.

$q \backslash \text{Ansatz}$	(39)	(40)	(41)
2	24	8	6
4	12	10	6
8	24	14	8
16	48	20	12
32	64	28	16
64	—	48	24

We note that $L_{\min,\min}$ increases with increasing q for all *Ansätze* that we use. The only exception is the *Ansatz* (39), where we have a minimum at $q = 4$. Going to large values of q , the amplitudes of corrections increase rapidly. This can also be seen directly by looking at the results obtained by using the *Ansätze* (40), (41).

Estimating the systematic errors that are due to the uncertainty in $K_{1,c}$ and K_3^* , we follow the discussions of Secs. IV A and IV B. The corresponding errors are given in Table II. We note that the error due to the uncertainty in $K_{1,c}$ is roughly the same for all values of q , while the error due to the uncertainty in K_3^* increases with increasing q .

F. Fitting by the large q -expansion

Starting from Eq. (1) we fitted our data for Δ_q by using the *Ansätze*

$$\Delta_q = c_{3/2}[q^{3/2} - (q-1)^{3/2}] + c_{1/2}[q^{1/2} - (q-1)^{1/2}] \quad (44)$$

TABLE II. Our final estimate of Δ_q for all values of q we simulated at. In () we give the error assuming that we are exactly at the critical coupling and that the amplitude of leading corrections is exactly vanishing. $\delta K_{1,c}$ is the possible error due to the uncertainty of our estimate of $K_{1,c}$ and δK_3^* the possible error due to the uncertainty of our estimate of K_3^* . For a discussion see the text.

q	Δ_q	$\delta K_{1,c}$	δK_3^*
2	0.71722(10)	0.000008	0.000005
3	0.87203(8)	0.000008	0.00002
4	1.00370(8)	0.000006	0.00003
5	1.11992(9)	0.000006	0.00003
6	1.22519(10)	0.000006	0.00005
7	1.32220(10)	0.000006	0.00005
8	1.41268(13)	0.000006	0.00006
16	1.99345(15)	0.000009	0.00008
24	2.4399(3)	0.000009	0.00012
32	2.8166(4)	0.000009	0.00015
48	3.4482(5)	0.000008	0.00024
64	3.9810(8)	0.000008	0.00032

TABLE III. We fit our data for Δ_q by using the *Ansätze* (44), (45). $c_{3/2}$, $c_{1/2}$, and $c_{-1/2}$ are the coefficients of Eq. (1).

$q_{\min}$	<i>Ansatz</i>	$c_{3/2}$	$c_{1/2}$	$c_{-1/2}$	χ^2/DOF
2	(45)	0.331553(22)	0.2815(5)	0.0191(9)	0.62
3	(45)	0.331566(27)	0.2809(9)	0.0173(22)	0.60
4	(45)	0.331598(31)	0.2787(14)	0.0095(45)	0.12
5	(45)	0.331602(37)	0.2784(22)	0.0081(89)	0.13
3	(44)	0.331725(18)	0.27389(27)	—	7.75
4	(44)	0.331648(21)	0.27587(37)	—	0.66
5	(44)	0.331628(23)	0.27648(49)	—	0.23
6	(44)	0.331623(25)	0.27670(62)	—	0.21

and

$$\Delta_q = c_{3/2}[q^{3/2} - (q-1)^{3/2}] + c_{1/2}[q^{1/2} - (q-1)^{1/2}] + c_{-1/2}[q^{-1/2} - (q-1)^{-1/2}]. \quad (45)$$

Since the error is partially systematic and furthermore the systematic error might point in the same direction for all values of q , χ^2/DOF has to be taken with caution. The same holds for the errors of the parameters of the fit. In the fit we only used the error, assuming that we are exactly at the critical temperature and that leading corrections to scaling vanish exactly.

In Table III we have summarized results of fits by using the *Ansatz* (45), where all Δ_q with $q \geq q_{\min}$ are taken into account. We see that χ^2/DOF makes a jump going from $q_{\min} = 3$ to 4. χ^2/DOF for $q_{\min} = 4$ is very small. Likely this is due to the fact that the systematic error in our estimates of Δ_q points in the same direction for different values of q . We also note that the values of $c_{3/2}$, $c_{1/2}$ change only little going from $q_{\min} = 4$ to 5. It is very remarkable that the estimates of $c_{-1/2}$ are very small and that they are decreasing with increasing $q_{\min}$. For $q_{\min} = 5$ the estimate of $c_{-1/2}$ is compatible with zero. It seems plausible that $c_{-1/2}$ is very tiny or even exactly zero and deviations from Eq. (44) are due to higher powers or even decay exponentially in q .

TABLE IV. Estimates of D_q obtained by summing up Δ_q . The starting point is $D_1 = 0.51908(1)$ [18]. To compare with the constant in Eq. (1) we compute $D_q - c_{3/2}q^{3/2} - c_{1/2}q^{1/2}$.

q	D_q	$D_q - c_{3/2}q^{3/2} - c_{1/2}q^{1/2}$
1	0.51908(1)	-0.0890
2	1.23630(12)	-0.0927
3	2.10833(23)	-0.0937
4	3.11203(34)	-0.0939
5	4.23195(47)	-0.0940
6	5.45714(62)	-0.0940
7	6.77934(78)	-0.0940
8	8.19202(97)	-0.0939

Hence we performed fits by using the *Ansatz* (44). The results are summarized on the lower half of Table III. Here χ^2/DOF is still large for $q_{\min} = 3$. However, very small already for $q_{\min} = 5$. We take our final estimates of $c_{3/2}$ and $c_{1/2}$ from this fit. Going to larger $q_{\min}$ the statistical error increases, likely without gaining much on systematic errors.

$$c_{3/2} = 0.33163(9) \quad (46)$$

$$c_{1/2} = 0.2765(8) \quad (47)$$

The error is due to different sources. First we take the error given in Table III. Then, assuming that the error is purely systematic and points in the same direction for all values of q , we have redone the fit using $\Delta_q + \epsilon$, where ϵ is the error that is quoted in Table II in (). The error is then the difference between fitting $\Delta_q + \epsilon$ and Δ_q . Analogously, we proceeded for $\Delta_q + \delta K_{1,c}$ and $\Delta_q + \delta K_3^*$. For $c_{3/2}$ these errors are 0.000023, 0.000042, 0.000002, and 0.000019, respectively. In the case of $c_{1/2}$ we get 0.00049, 0.00024, 0.000008, and 0.00011, respectively. To get the errors in Eqs. (46) and (47), we just added up these four numbers. We regard these error estimates as cautious.

Next, we add up the estimates of Δ_q , where we start from $D_1 = 0.51908(1)$ [18]. We just sum up the errors, assuming that they are systematic. The results are given in Table IV. To compare with the constant -0.0937256 in Eq. (1) we compute $D_q - c_{3/2}q^{3/2} - c_{1/2}q^{1/2}$. The difference seems to converge rapidly with increasing q . For $q = 5, 6$, and 7 it takes the value -0.0940 which is very close to the constant that is predicted analytically. The small difference can be explained by the error on D_q and the coefficients $c_{3/2}$ and $c_{1/2}$. It is remarkable that the difference even for $q = 1$ is just about 0.005.

V. SUMMARY AND OUTLOOK

We have determined the RG exponents $Y_q = 3 - D_q$ of a $\mathbb{Z}_q$ invariant perturbation at the $O(2)$ invariant fixed point in three dimensions by using Monte Carlo

TABLE V. Estimates for the RG exponent $Y_q = 3 - D_q$ obtained directly for the given $q \leq 6$. For $q < 4$, there is an abundance of results in the literature that we do not report here. We focus on recent results obtained by using the CB method and Monte Carlo (MC) simulations of lattice models. Note that the asterisk in the case of Ref. [27] indicates that the error estimate is no rigorous bound. For comparison, we give results obtained by using the iterative method in Refs. [14,17] and here.

Ref.	Method\q	1	2	3	4	5	6
[29]	ϵ -exp.				-0.114(4)		
[29]	3D exp.				-0.103(8)		
[30]	FRG				-0.111(12)		
[31]	CB	2.480912(22)	1.76371(11)	0.8914(3*)			
[27]	CB				-0.11535(73*)		
[32]	MC		1.7639(11)	0.8915(20)	-0.108(6)		
[33]	MC				-0.114(2)	-1.27(1)	-2.55(6)
[34]	MC				-0.1118(10)		
[23]	MC	2.48095(4)					-2.43(6)
[18]	MC	2.48092(1)					
[14]	MC	2.484(3)	1.762(5)	0.884(6)	-0.128(6)	-1.265(6)	-2.509(7)
[17]	MC	2.4801(98)	1.761(1)	0.884(2)	-0.124(2)	-1.254(3)	-2.486(4)
this work	MC		1.76370(12)	0.89167(23)	-0.11203(34)	-1.23195(47)	-2.45714(62)

simulations of an improved model. In particular, we studied the XY model with next-to-next-to-nearest neighbor couplings in addition to nearest neighbor couplings at the point, where leading corrections to scaling approximately vanish [18]. We simulate the model by using the worm algorithm [15,16]. The worm algorithm has been used to simulate the XY model in several previous works. See, for example, [24,25]. It is straight forward to apply the algorithm to the model studied here.

We build on previous work [14,17] on this problem. The dimensions D_q are obtained in an iterative approach. Considerable improvement is achieved by using site weights [19] in the worm simulation. The gain obtained by using this modification is increasing with increasing q . In particular, this allows us to reach $q = 64$ at high precision. The authors of [14,17] simulated the standard XY model. By simulating an improved model, we get better control on systematic errors caused by corrections to scaling. Comparing the central results $c_{3/2} = 0.3371(28)$ and $c_{1/2} = 0.266(35)$, Ref. [14], and $c_{3/2} = 0.340(1)$ and $c_{1/2} = 0.23(2)$, Ref. [17], with our $c_{3/2} = 0.33163(9)$, Eq. (46), and $c_{1/2} = 0.2765(8)$, Eq. (46), shows that in particular in Ref. [17], systematic errors are underestimated. The same holds for the estimates of D_q as can be seen in Table V.

In Table V we compare our results with estimates of Y_q that were obtained directly for the given value of q , by using various methods. We see that in general the error increases rapidly with increasing q . For $q \leq 3$ our results nicely agree with those obtained in Ref. [31] by using the conformal bootstrap method. The deviation for $q = 4$ between our estimate and that of Ref. [27] is about three times the sum of the errors that are quoted. On the other hand, we confirm the estimate of Ref. [34]. The estimates obtained by using the 5-loop ϵ -expansion and the 6-loop

expansion in three dimensions fixed are consistent with our result, but the error is clearly larger than ours. The same holds for Ref. [30], where the functional renormalization group (FRG) is employed. In the case of the Monte Carlo study of a lattice model [33], the estimate for $q = 4$ is consistent with ours, but too small for $q = 5$ and 6.

It had been noted before that Eq. (1) fits D_q down to surprisingly small values of q . This observation is further strengthened by our accurate estimates of D_q . It is remarkable that in our fits we can do without $O(q^{-1/2})$ for $q \gtrsim 4$.

ACKNOWLEDGMENTS

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under the Grant No. HA 3150/5-4.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX: RATIO OF PARTITION FUNCTIONS

The ratio of partition functions Z_a/Z_p has been a cornerstone in our finite size scaling analysis since Ref. [35]. Here, Z_a is the partition function of a system with antiperiodic boundary conditions in one of the three directions and the partition function Z_p of a system with periodic boundary conditions in all directions. Antiperiodic boundary conditions in 0-direction are obtained by

changing the sign of the term $\vec{s}_x \cdot \vec{s}_y$ of the Hamiltonian for nearest neighbor pairs, and here third nearest neighbor pairs, where $x_0 = L - 1$ and $y_0 = 0$. It turns out that it is straight forward to determine Z_a/Z_p using the worm simulation. First we note that

$$I_k(K) = (-1)^k I_k(-K). \quad (\text{A1})$$

Let us define

$$K_{L-1,0} = \sum_{\langle xy \rangle, x_0=L-1, y_0=0} k_{\langle xy \rangle} + \sum_{[xy], x_0=L-1, y_0=0} k_{[xy]}. \quad (\text{A2})$$

For a given configuration $\{k\}$, switching from periodic to antiperiodic boundary conditions, the product

$$W(\{k\}) = \left[\prod_{\langle xy \rangle} I_{k_{\langle xy \rangle}}(K_1) \right] \left[\prod_{[xy]} I_{k_{[xy]}}(K_3) \right] \quad (\text{A3})$$

changes sign if $K_{L-1,0}$ is odd. We can write the partition functions, without insertion of head and tail, as

$$Z_p = \sum_{\{k\}'} W(\{k\}'), \quad Z_a = \sum_{\{k\}'} W(\{k\}') (-1)^{K_{L-1,0}}, \quad (\text{A4})$$

where $\{k\}'$ are configurations that satisfy the constraint Eq. (10). Hence

$$\frac{Z_a}{Z_p} = \frac{\sum_{\{k\}'} W(\{k\}') (-1)^{K_{L-1,0}}}{\sum_{\{k\}'} W(\{k\}')} = \langle (-1)^{K_{L-1,0}} \rangle_p. \quad (\text{A5})$$

This means that the ratio of partition functions is just the expectation value of $(-1)^{K_{L-1,0}}$ for periodic boundary conditions. One should note that only loops with non-vanishing winding number contribute to $K_{L-1,0}$. Furthermore, $K_{L-1,0} = K_{0,1} = K_{1,2}, \dots$. It is straight forward to compute derivatives of Z_a/Z_p with respect to K_1 and K_3 . We have implemented the first and second derivative of Z_a/Z_p with respect to K_1 in our program. Furthermore we have implemented the internal energy and the magnetic susceptibility. Using $w(x, y) = 1$, without inserting charge, we performed high statistics simulations for smaller lattice sizes. The results are consistent with those of Ref. [18], which were obtained by using the cluster algorithm [36,37]. Benchmarked on the statistical error of Z_a/Z_p for example, the performance of the worm and cluster algorithm are similar.

-
- [1] K. G. Wilson and J. Kogut, The renormalization group and the ϵ -expansion, *Phys. Rep.* **12**, 75 (1974).
 - [2] M. E. Fisher, The renormalization group in the theory of critical behavior, *Rev. Mod. Phys.* **46**, 597 (1974); **47**, 543 (E) (1975).
 - [3] John Cardy, *Scaling and Renormalization in Statistical Physics*, Series: Cambridge Lecture Notes in Physics (No. 5) (Cambridge University Press, Cambridge, England, 1996).
 - [4] M. E. Fisher, Renormalization group theory: Its basis and formulation in statistical physics, *Rev. Mod. Phys.* **70**, 653 (1998).
 - [5] A. Pelissetto and E. Vicari, Critical phenomena and renormalization-group theory, *Phys. Rep.* **368**, 549 (2002).
 - [6] D. Poland, S. Rychkov, and A. Vichi, The conformal bootstrap: Theory, numerical techniques, and applications, *Rev. Mod. Phys.* **91**, 015002 (2019).
 - [7] Slava Rychkov and Ning Su, New developments in the numerical conformal bootstrap, *Rev. Mod. Phys.* **96**, 045004 (2024).
 - [8] L. F. Alday and J. M. Maldacena, Comments on operators with large spin, *J. High Energy Phys.* **11** (2007) 019.
 - [9] A. L. Fitzpatrick, J. Kaplan, D. Poland, and D. Simmons-Duffin, The analytic bootstrap and AdS superhorizon locality, *J. High Energy Phys.* **12** (2013) 004.
 - [10] Z. Komargodski and A. Zhiboedov, Convexity and liberation at large spin, *J. High Energy Phys.* **11** (2013) 140.
 - [11] S. Hellerman, D. Orlando, S. Reffert, and M. Watanabe, On the CFT operator spectrum at large global charge, *J. High Energy Phys.* **12** (2015) 071.
 - [12] A. Monin, D. Pirtskhalava, R. Rattazzi, and F. K. Seibold, Semiclassics, Goldstone bosons and CFT data, *J. High Energy Phys.* **06** (2017) 011.
 - [13] Luis Alvarez-Gaume, Domenico Orlando, and Susanne Reffert, Selected topics in the large quantum number expansion, *Phys. Rep.* **933**, 1 (2021).
 - [14] Debasish Banerjee, Shailesh Chandrasekharan, and Domenico Orlando, Conformal dimensions via large charge expansion, *Phys. Rev. Lett.* **120**, 061603 (2018).
 - [15] N. V. Prokof'ev, B. V. Svistunov, and I. S. Tupitsyn, Worm algorithm in quantum Monte Carlo simulations, *Phys. Lett. A* **238**, 253 (1998).
 - [16] N. Prokof'ev and B. Svistunov, Worm algorithms for classical statistical models, *Phys. Rev. Lett.* **87**, 160601 (2001).
 - [17] Gabriel Cuomo, J. M. Viana Parente Lopes, José Matos, Júlio Oliveira, and João Penedones, Numerical tests of the large charge expansion, *J. High Energy Phys.* **05** (2024) 161.
 - [18] M. Hasenbusch, Eliminating leading and subleading corrections to scaling in the three-dimensional XY universality class, *Phys. Rev. B* **112**, 184512 (2025).
 - [19] Ulli Wolff, Simulating the all-order strong coupling expansion I: Ising model demo, *Nucl. Phys.* **B810**, 491 (2009).

- [20] M. Hasenbusch and T. Török, High precision Monte Carlo study of the 3D XY-universality class, *J. Phys. A* **32**, 6361 (1999).
- [21] M. Campostrini, M. Hasenbusch, A. Pelissetto, P. Rossi, and E. Vicari, Critical behavior of the three-dimensional XY universality class, *Phys. Rev. B* **63**, 214503 (2001).
- [22] M. Campostrini, M. Hasenbusch, A. Pelissetto, and E. Vicari, The critical exponents of the superfluid transition in He-4, *Phys. Rev. B* **74**, 144506 (2006).
- [23] M. Hasenbusch, Monte Carlo study of an improved clock model in three dimensions, *Phys. Rev. B* **100**, 224517 (2019).
- [24] Debasish Banerjee and Shailesh Chandrasekharan, Finite size effects in the presence of a chemical potential: A study in the classical nonlinear $O(2)$ sigma-model, *Phys. Rev. D* **81**, 125007 (2010).
- [25] Wanwan Xu, Yanan Sun, Jian-Ping Lv, and Youjin Deng, High-precision Monte Carlo study of several models in the three-dimensional $U(1)$ universality class, *Phys. Rev. B* **100**, 064525 (2019).
- [26] S. Meneses, J. Penedones, S. Rychkov, J. M. Viana Parente Lopes, and P. Yvernaye, A structural test for the conformal invariance of the critical 3d Ising model, *J. High Energy Phys.* **04** (2019) 115.
- [27] Junyu Liu, David Meltzer, David Poland, and David Simmons-Duffin, The Lorentzian inversion formula and the spectrum of the 3d $O(2)$ CFT, *J. High Energy Phys.* **09** (2020) 115.
- [28] Junyu Liu and David Simmons-Duffin (private communication), communicated the numerical value $\omega_{NR} = 2.02548(41)$.
- [29] J. M. Carmona, A. Pelissetto, and E. Vicari, N -component Ginzburg-Landau Hamiltonian with cubic anisotropy: A six-loop study, *Phys. Rev. B* **61**, 15136 (2000).
- [30] Andrzej Chlebicki, Carlos A. Sánchez-Villalobos, Paweł Jakubczyk, and Nicolás Wschebor, $\mathbb{Z}_4$ -symmetric perturbations to the XY model from functional renormalization, *Phys. Rev. E* **106**, 064135 (2022).
- [31] S. M. Chester, W. Landry, J. Liu, D. Poland, D. Simmons-Duffin, N. Su, and A. Vichi, Carving out OPE space and precise $O(2)$ model critical exponents, *J. High Energy Phys.* **06** (2020) 142.
- [32] M. Hasenbusch and E. Vicari, Anisotropic perturbations in three-dimensional $O(N)$ -symmetric vector models, *Phys. Rev. B* **84**, 125136 (2011).
- [33] Hui Shao, Wenan Guo, and Anders W. Sandvik, Monte Carlo renormalization flows in the space of relevant and irrelevant operators: Application to three-dimensional clock models, *Phys. Rev. Lett.* **124**, 080602 (2020).
- [34] M. Hasenbusch, Monte Carlo study of the $O(2)$ -invariant ϕ^4 theory with a cubic perturbation in three dimensions, [arXiv:2510.19473](https://arxiv.org/abs/2510.19473).
- [35] M. Hasenbusch, K. Pinn, and S. Vinti, Critical exponents of the 3D Ising universality class from finite size scaling with standard and improved actions, *Phys. Rev. B* **59**, 11471 (1999).
- [36] Robert H. Swendsen and Jian-Sheng Wang, Nonuniversal critical dynamics in Monte Carlo simulations, *Phys. Rev. Lett.* **58**, 86 (1987).
- [37] U. Wolff, Collective Monte Carlo updating for spin systems, *Phys. Rev. Lett.* **62**, 361 (1989).