

Wilson loops with neural networks

Verena Bellscheidt^{1,*} Nora Brambilla^{2,3,4,†} Andreas S. Kronfeld^{5,3,‡} and Julian Mayer-Steedte^{2,4,§}

(TUMQCD Collaboration)

¹*Center for Theoretical Physics—A Leinweber Institute, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*

²*Department of Physics, School of Natural Sciences, Technische Universität München, James-Franck-Strasse 1, 85748 Garching, Germany*

³*Institute for Advanced Study, Technische Universität München, Lichtenbergstrasse 2a, 85748 Garching, Germany*

⁴*Munich Data Science Institute, Technische Universität München, Walther-von-Dyck-Strasse 10, 85748 Garching, Germany*

⁵*Theory Division, Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, Illinois 60510, USA*



(Received 9 February 2026; accepted 1 April 2026; published 1 May 2026)

Wilson loops are essential objects in QCD and have been pivotal in scale setting and demonstrating confinement. Various generalizations are crucial for computations needed in effective field theories. In lattice gauge theory, Wilson loop calculations face challenges, including excited-state contamination at short times and the signal-to-noise ratio issue at longer times. To address these problems, we develop a new method by using neural networks to parametrize interpolators for the static quark-antiquark pair. We construct gauge-equivariant layers for the network and train it to find the ground state of the system. The trained network itself is then treated as our new observable for the inference. Our results demonstrate a significant improvement in the signal compared to traditional Wilson loops, performing as well as Coulomb-gauge Wilson-line correlators while maintaining gauge invariance. Additionally, we present an example where the optimized ground state is used to measure the static force directly, as well as another example combining this method with the multilevel algorithm. Finally, we extend the formalism to find excited-state interpolators for static quark-antiquark systems. To our knowledge, this work is the first study of neural networks with a physically motivated loss function for Wilson loops.

DOI: [10.1103/qjip-qw7x](https://doi.org/10.1103/qjip-qw7x)

I. INTRODUCTION

The Wilson loop [1] is a widely used tool in mathematical and computational studies of gauge theories. In particular, in confining theories, such as quantum chromodynamics (QCD), Wilson loops of various shapes yield information about the energy carried by gluons in the presence of static sources. Energies are, of course, spectral quantities, which depend crucially on some properties of the loop(s) while being independent of other features.

This circumstance can be illustrated by a rectangular Wilson loop of spatial separation $\mathbf{r}$ and temporal extent t . For large (Euclidean) t , a spectral decomposition shows the Wilson loop to be a sum of terms of the form

$$\sum_n |c_n|^2 e^{-tE_n(\mathbf{r})}, \quad (1)$$

where the “static energies” $E_n(\mathbf{r})$ depend on the separation (usually just on the distance $r = |\mathbf{r}|$) and the representation in which the Wilson loop is evaluated. In the lattice-gauge-theory literature and elsewhere, the static energies are usually referred to as “static potentials.” The leading static energy $E_0(r)$ is the quarkonium static potential.¹ Higher

*Contact author: verena_b@mit.edu

†Contact author: nora.brambilla@ph.tum.de

‡Contact author: ask@fnal.gov

§Contact author: julian.mayer-steudte@tum.de

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) 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.

¹However, in the perturbative regime of small r , the potential suffers from infrared divergences [2] that are canceled in the static energies by contributions from ultrasoft gluons that can be identified in a multipole expansion [3,4]. Nonperturbatively, the static energy coincides with the static potential.

excitations of the static energies $E_n(r)$ correspond to the static potentials of hybrid states (with valence gluons) and, in the presence of light quarks, of tetraquark states [4–6].

As spectral quantities, the energies in Eq. (1) remain the same if the spatial parts of the Wilson loop are replaced by any spatial path connecting the same end points. Instead, the coefficients c_n depend on the spatial paths. In fact, it is possible and advantageous to design a combination of paths to isolate a state of particular interest.

For example, in lattice gauge theory, a common strategy is to fix the lattice gauge field to Coulomb gauge and then take the gauge-field average of Wilson lines; see, for example, Ref. [7]. Gauge averaging selects out combinations of paths that are gauge invariant, so the original gauge-invariant static energies are obtained. The static energies are presumably unaffected by the Gribov ambiguity of the Coulomb gauge. That said, experience [7] has shown that the best practice is to use the same algorithm with the same tolerances and stopping conditions for all configurations in an ensemble. Otherwise, somewhat different combinations of paths are implicitly chosen.

Such problems can be avoided with an explicitly gauge-invariant methodology that smears the spatial path, thereby maintaining the main virtue of the Coulomb gauge. In this paper, we use neural networks to generate an optimal set of paths. We develop and test the neural network on the lowest-lying static energy, finding that the neural network performs as well as the Coulomb gauge over a wide range of distances. We then adapt the neural network to optimize for the first excitation, which has different rotational quantum numbers. This neural network finds two degenerate levels—as expected.

A further advantage of the neural-network approach is that it can be applied to more complicated Wilson loops, once the basic setup has been established. Here, we proceed to develop and test a neural network for the *static force*, which is the spatial derivative of the static energy. We compute it not by an explicit discrete difference but by the insertion of plaquettes along the temporal Wilson lines [8–10]. Many other examples of modified Wilson loops [6,11–21] should be amenable to the neural-network technique, which we leave for future work.

While earlier works have applied neural networks to Wilson loops [22,23], this is—to our knowledge—the first to introduce a physically motivated loss function. Complementary machine-learning-based approaches to improve the operators and the signal are contour deformation of the path integral [24–26] and renormalization-group-improved actions [27]. For reviews of further work on machine learning in lattice gauge theory, see Refs. [28,29].

The rest of this paper is organized as follows. Section II presents the notation and theoretical background for Wilson loop studies in lattice gauge theory. The building blocks for neural networks are discussed in Sec. III. Results for the

lowest-lying static energy, a matrix element for the static force, and excited states are presented in Sec. IV. Section V concludes with a few remarks on future applications.

II. THEORETICAL BACKGROUND

A. Wilson loops

A rectangular, on-axis Wilson loop is defined in terms of link variables as

$$W_{r \times t} = S(\mathbf{x}, \mathbf{x} + \mathbf{r}, 0) T(0, t, \mathbf{x} + \mathbf{r}) S(\mathbf{x} + \mathbf{r}, \mathbf{x}, t) T(t, 0, \mathbf{x}), \quad (2)$$

where the factors are

$$S(\mathbf{x}, \mathbf{x} + \mathbf{r}, t) = \prod_{k=0}^{r-1} U_r(\mathbf{x} + k\hat{\mathbf{e}}_r, t), \quad (3)$$

$$S(\mathbf{x} + \mathbf{r}, \mathbf{x}, t) = S(\mathbf{x}, \mathbf{x} + \mathbf{r}, t)^\dagger, \quad (4)$$

$$T(0, t, \mathbf{x}) = \prod_{k=0}^{t-1} U_4(\mathbf{x}, k), \quad (5)$$

$$T(t, 0, \mathbf{x}) = T(0, t, \mathbf{x})^\dagger, \quad (6)$$

where $\hat{\mathbf{e}}_r$ is a unit vector in the direction of $\mathbf{r}$ and $U_\mu(\mathbf{x}) = U_\mu(\mathbf{x}, t)$ are the gauge links in direction $\mu = 1, 2, 3, 4$ (4 is the Euclidean time direction) at lattice site $\mathbf{x} = (\mathbf{x}, t)$, $\mathbf{x} = (x_1, x_2, x_3)$. The Wilson loop starts and ends at $\mathbf{x}$ in the time slice $t = 0$ with residing in the $\hat{\mathbf{e}}_r$ - $\hat{\mathbf{e}}_4$ plane with spatial and temporal extension $\mathbf{r}$ and t , respectively. Here, and for the rest of the paper, $\mathbf{r}$ and t are stated in lattice units, i.e., $t_{\text{ph}} = ta$ and $r_{\text{ph}} = ra$, where a is the lattice spacing.

A gauge-invariant quantity is obtained by taking the trace of the Wilson loop. We denote the gauge-field average as $\langle \text{tr} W_{r \times t} \rangle$. In a numerical simulation, it is approximated with a finite lattice ensemble as

$$\langle \text{tr} W_{r \times t} \rangle \approx \frac{1}{N_{\text{conf}}} \sum_{i=1}^{N_{\text{conf}}} \text{tr} W_{r \times t}^{(i)} \quad (7)$$

where the (i) superscript indicates the measurement of the observable on the i th configuration. Additionally, averaging is performed over all origins of the loop and all three spatial directions for $\mathbf{e}_r$.

The trace yields a globally gauge-invariant object, but individual parts are gauge variant. For example, imposing a gauge transformation $G(\mathbf{x})$, such that the gauge field transforms as

$$U_\mu(\mathbf{x}) \rightarrow G(\mathbf{x}) U_\mu(\mathbf{x}) G^\dagger(\mathbf{x} + \hat{\mathbf{e}}_\mu), \quad (8)$$

transforms the spatial Wilson lines accordingly to

$$S(\mathbf{x}, \mathbf{x} + \mathbf{r}, t) \rightarrow G(\mathbf{x}, t) S(\mathbf{x}, \mathbf{x} + \mathbf{r}, t) G^\dagger(\mathbf{x} + \mathbf{r}, t), \quad (9)$$

which corresponds to the gauge property of a static quark-antiquark pair located at $(\mathbf{x}, t)$ and $(\mathbf{x} + \mathbf{r}, t)$.

The Wilson loop is interpreted as a two-point function of a mesonic state, $|S_r\rangle$, composed of two static quarks separated by r where $S(\mathbf{x} + \mathbf{r}, \mathbf{x}, t) \cong S_r(t)$ is its interpolator. Therefore, the Wilson loops represent transfer matrix elements of $S(\mathbf{x} + \mathbf{r}, \mathbf{x}, t)$ as

$$\langle \text{tr } W_{r \times t} \rangle \cong \langle S_r(0) S_r^\dagger(t) \rangle = \langle S_r | T^t | S_r \rangle \quad (10)$$

with the transfer matrix $T = e^{-aH}$. Assuming positivity, or at least physical positivity [30], of the transfer matrix, the spectral decomposition of the Wilson loop is

$$\langle \text{tr } W_{r \times t} \rangle = \sum_{n=0}^{\infty} |c_n|^2 e^{-taE_n(r)}, \quad (11)$$

where $c_n = \langle n | S_r \rangle$ is the overlap of state n with the spatial Wilson line and $E_n(r)$ the energy (in physical units, hence the lattice spacing a) of the n th state. The ground-state energy $E_0(r)$, i.e., $E_0(r) < E_n(r)$ for $n > 0$, defines the static energy of the heavy quark-antiquark pair, and it can be used in a Schrödinger equation to calculate the energy levels of quarkonium, hybrids, or tetraquarks [4–6].

B. Generalized Wilson loops

The Wilson loop Eq. (2) is one single, closed loop of link variables; hence, $W_{r \times t} \in \text{SU}(3)$. The definition of the Wilson loop generalizes by considering Wilson loops with field-strength component insertions in the temporal or spatial Wilson lines, replacing the spatial Wilson lines with sums of paths with different shapes, inserting light quark pairs in the spatial Wilson lines, or a combination of those possibilities, or both. For example, the spatial Wilson lines can be replaced with sums over paths of varying shapes, which can then project better onto specific states $|n\rangle$.

Inserting one chromoelectric field component on the temporal Wilson line yields the ground-state matrix element of the chromoelectric field component, which is a measure for the static force between the static quark-antiquark pair [8–10]. Multiple insertions of chromoelectric and chromomagnetic field components in the temporal Wilson lines yield the spin and relativistic correction terms to the quarkonium static potential [5, 11–16, 31].

Insertions of chromomagnetic and chromoelectric field components and their combinations in the spatial Wilson line produce interpolators with a specific quantum number, creating the hybrid states. These interpolators are not unique and are replaced by sums of differently shaped paths that still obey the correct quantum number but have better overlap with the hybrid state in question. These ideas have been extensively studied in the literature [18, 20, 32–35]. Insertions of chromoelectric and chromomagnetic field components in the temporal Wilson lines together with insertion of appropriate interpolators in the

spatial lines yield spin and relativistic correction terms to the hybrids and tetraquark static potentials [6, 17, 19].

C. Neural networks for Wilson loops

In this work, we study the description of Wilson loops with a neural network designed to parametrize optimal interpolators with respect to the ground-state overlap. We propose replacing the spatial Wilson line by a neural network $\tilde{S}$ with the same end points as S . At the same time, we keep the temporal Wilson lines fixed. The neural network $\tilde{S}$ depends on a set of parameters $\{w^{(n)}\}$, where n iterates through the layers of the network. By optimizing these parameters, we maximize the ground-state overlap $|c_0|^2$ while minimizing the excited-state contributions $|c_n|^2$, $n > 0$. By leaving the temporal Wilson lines untouched, we preserve the spectrum $E_n(r)$.

To retain gauge invariance, the gauge-transformation property of $\tilde{S}$ has to be the same as for S ; i.e., under a gauge transformation $G(x)$, it must transform as

$$\tilde{S}(\mathbf{x}, \mathbf{x} + \mathbf{r}, t) \rightarrow G(\mathbf{x}, t) \tilde{S}(\mathbf{x}, \mathbf{x} + \mathbf{r}, t) G^\dagger(\mathbf{x} + \mathbf{r}, t). \quad (12)$$

We achieve this by starting with an initial set of paths that obey the correct gauge property and recombining them by iterating through layers of a neural network.

A useful network has to fulfill several requirements, stemming from gauge invariance and practical considerations. First, the layers must be gauge equivariant, meaning that the output elements conform to the same gauge property as the input elements. Second, the layers must exhibit high expressivity in representing the underlying gluon dynamics, and the network must systematically develop complicated structures. Third, the input elements and layers should be simple to implement and to generalize to different separations r and directions $\mathbf{e}_r$, thereby avoiding the explicit construction of complicatedly shaped objects. The new, generalized Wilson loop $\tilde{W}_{r \times t}(\{w^{(n)}\})$ is now parametrized by a set of parameters $\{w^{(n)}\}$ and can be optimized with respect to a loss function L . The fourth and final requirement is that the derivative $\partial L / \partial w^{(n)}$ must exist. These four requirements will be resolved in this study by developing and implementing gauge-equivariant layers, imposing a loss function, training the neural network, and demonstrating its capability.

III. BUILDING BLOCKS

This section covers the building blocks of our neural-network approach. We cover the definition of gauge-equivariant layers, the general neural-network architecture, and the process of determining the loss function. Those technical details are crucial for establishing the foundation for further applications of the neural-network method.

A. Layers for general states

To parametrize general static quark-antiquark states, we employ a feedforward neural network whose layers iteratively recombine open paths sharing the gauge transformation law of the straight spatial Wilson line. As emphasized in Sec. II C, the guiding principle is gauge equivariance: If each layer maps inputs with the static-line gauge property to outputs with the same property, the full network inherits this equivariance. Specifically, we restrict ourselves to algebraic operations—most notably linear superpositions and simple local compositions—that preserve the end points of the paths and thus their gauge behavior, ensuring that the network output remains a valid interpolator for the static pair. The trainable weights are chosen to be identical for all lattice sites and all orientations of $\mathbf{r}$, thereby encoding translational and rotational (on the cubic lattice) invariance directly in the architecture. With this approach, the network realizes a systematically improvable superposition of paths of increasing geometric complexity while remaining straightforward to implement. The following subsections introduce the corresponding gauge-equivariant recombinations that serve as our building blocks. Similar layers and networks were developed in Refs. [22,36,37].

The initial assumption is that all open paths of ordered link variables starting and ending at the same point share the same gauge property, and we can define gauge-equivariant recombinations of them. A recurrent combination is the sum of two objects. Let us assume ϕ_1 and ϕ_2 transform identically as

$$\phi_1 \rightarrow G_1 \phi_1 G_2^\dagger, \quad (13)$$

$$\phi_2 \rightarrow G_1 \phi_2 G_2^\dagger \quad (14)$$

with some gauge transformations G_1 and G_2 with $G_i \in \text{SU}(3)$ and $\phi_i \in \mathbb{C}^{3 \times 3}$, the set of complex 3×3 matrices. The sum of both objects preserves the gauge transformation

$$\phi_s = \phi_1 + \phi_2 \rightarrow G_1 \phi_1 G_2^\dagger + G_1 \phi_2 G_2^\dagger \quad (15)$$

$$= G_1 (\phi_1 + \phi_2) G_2^\dagger = G_1 \phi_s G_2^\dagger, \quad (16)$$

where the final object is a superposition of different paths with different shapes or any more complicated object, while the start and end point are preserved. Gauge-equivariant layers rely on this property, with each term in the sum assigned a unique weight that represents the layer's parameters. These weights are independent of the field configuration, specific lattice site, and the orientation of the Wilson loop. This independence ensures the gauge, translational, and rotational invariance of our setup.

The focus in this study lies on on-axis separations between $\mathbf{x}$ and $\mathbf{y}$, i.e., $\mathbf{y} = \mathbf{x} + r\hat{\mathbf{e}}_r$, where $\hat{\mathbf{e}}_r$ is the direction

of separation and $(\mathbf{x}, t)$ a generic lattice site. However, the procedure can be generalized to off-axis separations as well, which we postpone to future studies.

In the case of on-axis separations, the simplest gauge-variant object is the straight Wilson line connecting $\mathbf{x}$ and $\mathbf{y}$, which we denote as $\phi^{(0)}(\mathbf{x})$; in the notation of Eq. (3), $\phi^{(0)}(\mathbf{x}) = S(\mathbf{x}, \mathbf{x} + \mathbf{r}, t)$. The next level of straightforward gauge-equivariant modifications involves inserting plaquettes at one of the lattice sites along the straight Wilson line. There are 24 possible oriented plaquettes in the spatial subvolume containing each site.

A modified Wilson line is given by the expression

$$\begin{aligned} \phi_{l,i,j}(\mathbf{x}) &= [\Pi_{k=0}^l U_r(\mathbf{x} + k\hat{\mathbf{e}}_r; t)] U_{i,j}(\mathbf{x} + l\hat{\mathbf{e}}_r; t) \\ &\times [\Pi_{k=l}^{r-1} U_r(\mathbf{x} + k\hat{\mathbf{e}}_r; t)], \end{aligned} \quad (17)$$

where $U_{i,j}(\mathbf{x}) = U_{i,j}(\mathbf{x}; t)$ is an oriented plaquette in the i - j plane at site $\mathbf{x}$ defined as

$$U_{i,j}(\mathbf{x}) = U_i(\mathbf{x}) U_j(\mathbf{x} + \hat{\mathbf{e}}_i) U_{-i}(\mathbf{x} + \hat{\mathbf{e}}_i + \hat{\mathbf{e}}_j) U_{-j}(\mathbf{x} + \hat{\mathbf{e}}_j) \quad (18)$$

and the indices run as $l = 0, 1, \dots, r$ and $i, j \in \{1, 2, 3, -1, -2, -3\}$. At insertion locations $l \neq 0$, the four plaquette insertions with a link variable pointed in $-\hat{\mathbf{e}}_r$ direction are already accounted for by the plaquette insertions at $l - 1$. Hence, we obtain $24 + 16r$ independent plaquette insertions for a given distance r .

A multi-index i , running from 1 to $24 + 16r$, labels all plaquette insertions and is denoted as $\phi_i^{(0)}(\mathbf{x})$. To maintain consistency, we need to establish a convention for the order of insertions. This becomes crucial if the direction of separation changes even though the physics remains invariant. Therefore, we define the order of coordinates as $(1, 2, 3)$ for $\hat{\mathbf{e}}_r = \hat{\mathbf{e}}_1$, and we adjust the order of the coordinates for different directions (μ_1, μ_2, μ_3) such that the ϵ tensor is 1, i.e., $\epsilon_{\mu_1 \mu_2 \mu_3} = 1$. Consequently, for $\hat{\mathbf{e}}_r = \hat{\mathbf{e}}_2$ the order becomes $(2, 3, 1)$, and for $\hat{\mathbf{e}}_r = \hat{\mathbf{e}}_3$ it is $(3, 1, 2)$. This convention ensures a consistent ordering of $\phi_i^{(0)}(\mathbf{x})$, keeping the neural-network parameters unchanged when evaluating for a different direction. The plaquette-inserted Wilson lines serve as inputs to the neural network [indicated by the (0) superscript], and their recombination through the equivariant layers returns a more complex expression.

For the rest of this work, the subscript of $\phi_i^{(n)}$ ($\phi_i^{(n+1)}$) is an integer with $i = 1, \dots, N^{(n)}$ ($i = 1, \dots, N^{(n+1)}$) where $N^{(n)}$ ($N^{(n+1)}$) is also sometimes referred to as the number of channels. It is fixed only for $n = 0$ ($N^{(0)} = 24 + 16r$ plaquette insertions), while $n > 0$ generally indicates the n th layer. $N^{(n)}$ may vary between layers but must be 1 for the final layer if we consider only a single final state.

We now introduce several layers that are used in the neural networks designed below. A *linear layer* is a sum of input elements defined as

$$\phi_i^{(n+1)}(x) = \sum_j w_{ij}^{(n+1)} \phi_j^{(n)}(x) - b_i^{(n+1)} \phi^{(0)}(x), \quad (19)$$

with the weights $w^{(n+1)} \in \mathbb{K}^{N^{(n+1)} \times N^{(n)}}$ and $b^{(n+1)} \in \mathbb{K}^{N^{(n+1)}}$ for $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. The term with b_i serves as the bias term. In comparison to a regular linear layer, the bias term needs to be multiplied by the straight Wilson line to maintain gauge equivariance of the layer; this aspect allows us to identify it as the neutral element within our algebra. With Eq. (16) and assuming the input elements obey the required gauge property, this layer is gauge equivariant.

The derivative with respect to the weights of the layers exists and is analytic; hence, automatic differentiation computes it during the backward pass. However, if $\mathbb{K} = \mathbb{C}$, the derivatives with respect to the real and imaginary parts of the weights have to be performed independently.

The purpose of the linear layer is to produce superpositions of plaquette insertions within the Wilson lines, which are associated with cloverlike insertions of the field-strength components $F_{\mu\nu}$. Field-strength insertions create excited states characterized by a unique quantum number. Consequently, this layer enhances the expressiveness of excited states, which we aim to minimize in the final output.

Another motivation for this layer is to reduce the number of channels if we do not expect such a high number of degrees of freedom. For example, the initial $24 + 16r$ possible plaquette insertions consume a high amount of memory, but the actual objects are the field insertions. Hence, an initial linear layer may be used to reduce the number of channels before proceeding with the next layer.

A *bilinear layer* combines the input elements bilinearly; however, only the insertion of the straight Wilson line

within the bilinear combination maintains gauge equivariance. Therefore, we define the bilinear layer as

$$\begin{aligned} \phi_i^{(n+1)}(x) = & \sum_{j,k} w_{ijk}^{(n+1)} \phi_j^{(n)}(x) (\phi^{(0)}(x))^\dagger \phi_k^{(n)}(x) \\ & + \sum_j \tilde{w}_{ij}^{(n+1)} \phi_j^{(n)}(x) - b_i^{(n+1)} \phi^{(0)}(x) \end{aligned} \quad (20)$$

with the weights $w^{(n+1)} \in \mathbb{K}^{N^{(n+1)} \times N^{(n)} \times N^{(n)}}$, $\tilde{w}^{(n+1)} \in \mathbb{K}^{N^{(n+1)} \times N^{(n)}}$, and $b^{(n+1)} \in \mathbb{K}^{N^{(n+1)}}$ for $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. The second and third terms correspond to the linear layer and are optional. However, they preserve the contributions of the input elements, which is the case if the bilinear objects cover only partially the underlying problem. The overall motivation for this layer is, for example, that the output elements are proportional to multiple plaquette insertions when the initial elements consist of single plaquette insertions. Thus, this layer introduces structure along the separation axis.

The bilinear layer is gauge equivariant, as can be demonstrated by a straightforward calculation. Furthermore, $\phi_i^{(n+1)}$ is analytically differentiable with respect to its weights, and, hence, it is calculated in the backward process. However, the bilinear layer has the disadvantage of producing numerically unstable objects. The objects $\phi_i^{(n)}$ are not necessarily SU(3) matrices and might have very small or large numbers, which become even smaller or larger after applying the bilinear layer repeatedly. This may quickly exceed the float number range, leading to instability during optimization. Therefore, the use of the bilinear layer requires some care. Avoiding bilinear layers and using exponential activation functions might be a more stable and self-regulated approach.

A *convolutional layer* combines objects from neighboring lattice sites using parallel transport to maintain gauge equivariance. Hence, the convolutional layer is defined as

$$\phi_i^{(n+1)}(x) = \sum_j \sum_{\mu=0,\pm 1,\pm 2,\pm 3} w_{ij\mu}^{(n+1)} U_\mu(x) \phi_j^{(n)}(x + \hat{e}_\mu) U_{-\mu}(x + \hat{e}_\mu + \mathbf{r}) - b_i^{(n+1)} \phi^{(0)}(x) \quad (21)$$

$$\begin{aligned} = & \sum_j \left\{ \sum_{\mu=1,2,3} \left[\tilde{w}_{ij\mu}^{(n+1)} U_\mu(x) \phi_j^{(n)}(x + \hat{e}_\mu) U_\mu^\dagger(x + \mathbf{r}) + \hat{w}_{ij\mu}^{(n+1)} U_\mu^\dagger(x - \hat{e}_\mu) \phi_j^{(n)}(x - \hat{\mu}) U_\mu(x + \mathbf{r} - \hat{\mu}) \right] \right. \\ & \left. + w_{ij,0}^{(n+1)} \phi_j^{(n)}(x) \right\} - b_i^{(n+1)} \phi^{(0)}(x) \end{aligned} \quad (22)$$

with $\tilde{w}_{ij\mu}^{(n+1)} = w_{ij\mu}^{(n+1)}$ and $\hat{w}_{ij\mu}^{(n+1)} = w_{ij,-\mu}^{(n+1)}$, and $w^{(n+1)} \in \mathbb{K}^{N^{(n+1)} \times N^{(n)} \times 7}$, $b^{(n+1)} \in \mathbb{K}^{N^{(n+1)}}$ for $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$. Equation (21) defines the convolutional layer, while Eq. (22) is a small change in notation for the weights corresponding to the technical implementation. The last term

corresponds to the bias term and is optional. Note that in Eq. (21) the case $\mu = 0$ represents a shorthand notation for no convolution, which is explicitly denoted in the second line of Eq. (22). Thus, the term containing $w_{ij,0}^{(n+1)}$ generates a linear layer that preserves structures obtained in the previous layer.

The convolutional layer introduces structure orthogonal to the separation axis. Although it does not produce explicitly objects proportional to rectangular plaquette insertions, it produces objects with a prolonged extension perpendicular to the separation axis, and the output object is a superposition that might have rectangular plaquette insertions. The exact contribution of the individual objects is, however, governed by the tunable weights. Furthermore, the output objects can be interpreted as fat links obtained similarly in smearing procedures such as APE smearing [38]. The key difference is that the exact shape of the smearing is parametrized by the weights, while in APE smearing a specific choice is made by hand. As with the linear and bilinear layers, the convolutional layer is a parametrized gauge-equivariant recombination and is analytically differentiable with respect to the weights. Figure 1 illustrates examples of how the linear, bilinear, and convolutional layers create gauge-equivariant structure through iterative application.

An $SU(3)$ exponential activation is a parameter-free layer mapping any complex 3×3 matrix onto $SU(3)$. The technical implementation consists of two steps, which are outlined below. The first step projects out the traceless, anti-Hermitian part of $\phi_j^{(n)}(\phi^{(0)}(x))^\dagger$, an object that transforms locally at x , as

$$\begin{aligned} \tilde{\phi}_i^{(n)}(x) &= \phi_i^{(n)}(x)(\phi^{(0)}(x))^\dagger - \left(\phi_i^{(n)}(x)(\phi^{(0)}(x))^\dagger \right)^\dagger \\ &\quad - \frac{1}{3} \text{tr} \left[\phi_i^{(n)}(x)(\phi^{(0)}(x))^\dagger - \left(\phi_i^{(n)}(x)(\phi^{(0)}(x))^\dagger \right)^\dagger \right]. \end{aligned} \quad (23)$$

$\tilde{\phi}^{(n)}$ inherits the gauge property of $\phi_j^{(n)}(\phi^{(0)}(x))^\dagger$, and it is an element of $\mathfrak{su}(3)$. Hence, an exponentiation of this object returns an $SU(3)$ matrix. The second step multiplies the exponential with $\phi^{(0)}$,

$$\phi_i^{(n+1)}(x) = e^{\tilde{\phi}_i^{(n)}(x)} \phi^{(0)}(x). \quad (24)$$

The exponential of a locally gauge-transforming object remains a locally gauge-transforming object, which can be verified by representing the exponential as a power series. The final multiplication with $\phi^{(0)}$ preserves the gauge equivariance.

The $SU(3)$ exponential activation function enjoys two advantages. The first one includes combinations of the objects similar to bilinear, trilinear, and n -linear operations with decreasing weights while still producing $SU(3)$ matrices that are regulated. That means they have well-behaved numbers that do not encounter floating-point issues while capable of similar complexity. The second is based on the regularization property; any final state of the neural network has a clear norm, which we discuss below when constructing the loss function.

The matrix exponential is an analytic operation, so it is differentiable with respect to the dependent variables. We use the Fréchet derivative [39,40] implemented in the built-in routines of PYTORCH, which computes the derivative of a scalar and real loss function for the steepest descent with respect to real-valued parameters. This qualifies it for gradient descent methods, which optimize for a defined loss function.

In summary, the linear, bilinear, convolutional, and exponential activation layers provide a set of

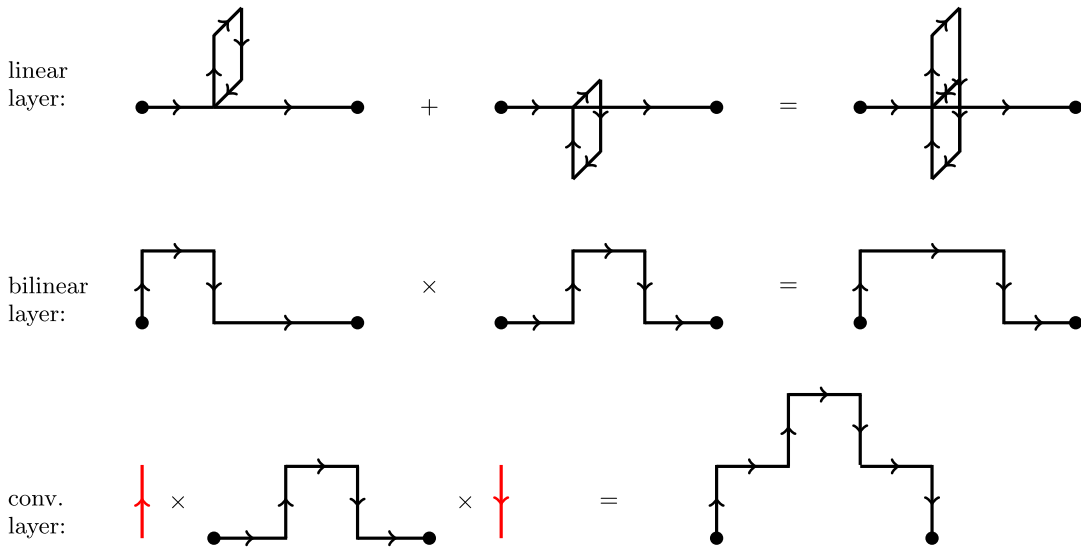


FIG. 1. Sketch of the actions of linear (top), bilinear (middle), and convolutional (bottom) layers, showing how gauge equivariance is built in. The black dots represent static quark-antiquark pairs, while the arrowed lines show the path connecting them. The linear layer creates clover-leaf-like insertions; the bilinear layer generates structure along the separation axis; and the convolutional layer creates structure orthogonal to the separation axis.

gauge-equivariant building blocks of a feedforward neural network. Starting with simple plaquette insertions that obey the correct gauge property, the output of the neural network exhibits the same gauge property and can be understood as a sum of differently shaped paths. The contributions of the individual paths are parametrized by the layer's weights and can be tuned such that the neural network optimizes a chosen objective function. The level of complexity is tuned by the number of hidden layers, the number of hidden channels, and the choice of the layers. These choices constitute a class of hyperparameters.

B. Neural-network architecture

Section III A defines gauge-equivariant layers which have the same gauge property as the straight Wilson line defined in Eq. (3). Here we use the layers to parametrize a static quark-antiquark state, replacing the spatial Wilson lines in Eq. (2). The spatial part corresponds to a single superposition of multiple paths connecting $\mathbf{x}$ to $\mathbf{x} + \mathbf{r}$; therefore, the number of channels of the final layer is equal to 1 (i.e., $N^{(n_{\text{final}})} = 1$) in the regular case. We might denote it with

$$\tilde{S}(\mathbf{x}, t) = \phi_1^{(n_{\text{final}})}(x) \quad (25)$$

and in Eq. (2) we replace both $S(x)$, at time slices 0 and t , with $\tilde{S}(x)$, keeping the straight temporal Wilson lines.

We write the traced neural-network parametrized generalization of the Wilson loop as $\langle \text{tr} \tilde{W}_{r \times t}(\{w^{(n)}\}) \rangle$, explicitly including the dependence on the weights, because they serve as the parameters of our variational *Ansatz*. For the rest of the study, the expression $\langle \text{tr} \tilde{W}_{r \times t} \rangle$ contains the dependence on the weights implicitly. The expectation value of the Wilson loop is real valued, but it might have a remnant imaginary part of the order of the floating-point precision. Therefore, it is safe to take the real part in numerical work.

The norm of a state is defined as $\langle \tilde{S} | \tilde{S} \rangle \equiv \langle \text{tr} \tilde{W}_{r \times 0} \rangle = \sum_x \langle \text{tr} \tilde{S}(x) \tilde{S}^\dagger(x) \rangle / V$ with the lattice volume V . If $\tilde{S}(x)$ is an $\text{SU}(3)$ matrix, the norm will always be equal to 3. In the case of a parametrized superposition of different paths, the explicit norm depends on the weights; however, the norm does not affect the spectral quantity of the observable, and it plays only a role in the numerical stability of the calculation.

To build a network, we can choose either a *flat* network with only a few hidden layers or a *deep* network with many hidden layers. It turns out that a flat network converges faster to a saturation point in the loss function, while a deep network converges slowly or not at all. In addition, many initial bilinear layers may encounter numerical problems with numerical zeros or not-a-numbers (NaNs) because they are related to high-degree polynomials of the lattice field numbers. On the other hand, a flat network might not

capture the full complexity of gluon dynamics. The dynamics have an effective range corresponding to the radius of the flux tube; each application of a convolutional layer increases the effective range of the network by one lattice spacing. Therefore, we need to find a balance between flat and deep network architectures. We employ progressive learning; namely, we begin with a flat network and add new layers iteratively in a controlled manner.

The general network construction procedure is as follows: In the first step, we set up an initial network. The first layer consists of a linear layer with $N^{(1)} = 16$ or $N^{(1)} = 14$.² Since we do not expect all initial plaquette insertions to be independent, this layer effectively reduces the number of elements, which saves memory on the hardware required for the backward call. After the initial layer, two to four different combinations of convolutional, bilinear, and exponential layers follow, where we keep the number of input and output elements constant and equal for the hidden layers. We initialized the weights randomly, such that the initial network remains close to the straight Wilson line.

In the second step, we optimize the network with respect to a loss function for a specified number of epochs, reaching a saturation point in the loss function. We add a new layer one before the final layer in the third step, initiated with random weights. To adjust the new weights, we freeze the previous weights and perform a few epochs only with the new weights before continuing with the full training. We repeat steps two and three until we reach a limit. These limits may be due to memory bounds or a loss of training stability. By storing all intermediate networks, we can select the optimal network for our purpose by inspecting the loss function as it evolves over the course of the number of epochs. This simple procedure works only if the number of input and output elements of the hidden layers is the same.

C. Loss function

To optimize the network towards a physically meaningful result, we need to motivate loss functions that aim for the physics we are interested in. In quantum many-body physics, a variational *Ansatz* $|\Psi(\{w^{(n)}\})\rangle$ parametrized by a set of weights $\{w^{(n)}\}$ is optimized to approach the ground state of the system by minimizing the energy expectation value, i.e.,

$$|\Psi_0\rangle \approx |\Psi(\{w^{(n)}\}_0)\rangle, \quad \{w^{(n)}\}_0 = \arg \min_{\{w^{(n)}\}} \frac{\langle \Psi(\{w^{(n)}\}) | H | \Psi(\{w^{(n)}\}) \rangle}{\langle \Psi(\{w^{(n)}\}) | \Psi(\{w^{(n)}\}) \rangle}. \quad (26)$$

²In the course of this work, we have not carried out a detailed study of the neural networks' dependence on $N^{(1)}$.

Neural network quantum states (NQS) are a class of variational *Ansätze* to approximate the ground state of a system by using a strategy to minimize Eq. (26); see, for example, [41]. However, an NQS provides only an approximation of the ground state, and further inspections of its correctness are needed. In our study, we aim to approximate the ground state of the static quark-antiquark system. In contrast to NQS, an approximation is sufficient, as the Euclidean time propagation converges to the ground state. In this case, a good ground-state approximation leads to stabler fitting at earlier Euclidean-time separations, which is crucial for precision calculations.

In lattice-gauge-theory computations, the energy expectation value $\langle \Psi(\{w\}) | H | \Psi(\{w\}) \rangle$ is not directly accessible. Instead, a lattice correlator provides the matrix elements of the transfer matrix in Eq. (10), with $T = e^{-aH}$. Since the transfer matrix is an exponential of the negative Hamiltonian, and this function is strictly monotonically decreasing, Eq. (26) can be reformulated to

$$|\tilde{S}_0\rangle \approx |\tilde{S}(\{w^{(n)}\}_0)\rangle, \quad \{w^{(n)}\}_0 = \arg \min_{\{w^{(n)}\}} \frac{\langle \tilde{S}(\{w^{(n)}\}, t) | \tilde{S}(\{w^{(n)}\}, 0) \rangle}{\langle \tilde{S}(\{w^{(n)}\}) | \tilde{S}(\{w^{(n)}\}) \rangle}. \quad (27)$$

We identify $\tilde{S}(\{w^{(n)}\}, t)$ as our neural-network Wilson line parametrized by the set $\{w^{(n)}\}$ at time slice t and $\langle \text{tr} \tilde{W}_{r \times t} \rangle \equiv \langle \tilde{S}(\{w^{(n)}\}, t) | \tilde{S}(\{w^{(n)}\}, 0) \rangle$. It follows that $\langle \text{tr} \tilde{W}_{r \times 0} \rangle \equiv \langle \tilde{S}(\{w^{(n)}\}) | \tilde{S}(\{w^{(n)}\}) \rangle$.

Equation (27) thus provides an objective function for Euclidean correlators which, when the parameters are optimized with respect to it, delivers an improvable approximation of the initial ground state. Two ambiguities remain, however, in the final formulation of the loss function. One is conceptual, since we can state independent conditions for each t , while the second is numerical, since $\langle \tilde{S}(\{w^{(n)}\}) | \tilde{S}(\{w^{(n)}\}) \rangle$ is not uniquely fixed.

We split the loss function for $\langle \text{tr} \tilde{W}_{r \times t} \rangle$ in a physical loss function L^{phys} which encodes Eq. (27) and its ambiguity in t and a regulator loss function L^{reg} which regulates the ambiguity of $\langle \tilde{S}(\{w^{(n)}\}) | \tilde{S}(\{w^{(n)}\}) \rangle$ and provides numerical stability while having no impact on the physical outcome. The final loss function is given by

$$L = L^{\text{phys}} + L^{\text{reg}} \quad (28)$$

with the individual terms being given by

$$L^{\text{phys}} = - \sum_{t=1}^{t_{\text{max}}} W_t \frac{\langle \text{tr} \tilde{W}_{r \times t} \rangle}{\langle \text{tr} \tilde{W}_{r \times 0} \rangle}, \quad (29)$$

$$L^{\text{reg}} = W(\langle \text{tr} \tilde{W}_{r \times 0} \rangle - N)^2, \quad (30)$$

$$\sum_{t=1}^{t_{\text{max}}} W_t = t_{\text{max}}, \quad (31)$$

where t_{max} , W_t , N , and W are hyperparameters, to be discussed further below. The minus sign in L^{phys} recasts the maximizing condition, Eq. (27), as a minimizing problem, as is standard in machine learning. The hyperparameters t_{max} , W_t , and N play the following roles: t_{max} sets the maximum time difference, once the signal at larger times becomes statistically weak; W_t is used to fine-tune the optimization window, because excited-state contamination is more dominant at small t , while noise dominates at larger t ; N sets the target norm; and W adjusts the importance of the norm condition. Equation (31) is a simple normalization convention based on the case where $W_t = 1$, a uniformly weighted window. Given this formulation of the loss function, we see that L^{phys} drives the network toward its ground state, while L^{reg} prevents the network from running into numerical instabilities emerging in numerical zeros or NaNs.

The derivation of the loss function can also be obtained directly from the spectral representation in Eq. (11). If $\tilde{c}_n$ [for $c_n \rightarrow \tilde{c}_n$ in Eq. (11), while the E_n remain the same] are the overlap coefficients of the neural-network state with the n th state, the optimal case is $\tilde{c}_0 \approx 1$ and $\tilde{c}_{n>0} \approx 0$. However, $\langle \text{tr} \tilde{W}_{r \times t} \rangle$ does not provide direct access to $\tilde{c}_n$, so the loss function cannot directly optimize for the ideal case $\tilde{c}_0^{\text{ideal}} = 1$, $\tilde{c}_{n>0}^{\text{ideal}} = 0$. Instead, it is necessary to argue through the correlator itself, as we do in the following.

We start by inspecting the ideal case, which is only the ground state without excited-state contamination, given by a single exponential decay as

$$\frac{\langle \text{tr} \tilde{W}_{r \times t}^{\text{ideal}} \rangle}{\langle \text{tr} \tilde{W}_{r \times 0}^{\text{ideal}} \rangle} = e^{-taE_0(r)}, \quad (32)$$

where, in general, $\tilde{c}_0^{\text{ideal}} \neq 1$ but we focus on $\tilde{c}_0^{\text{ideal}} = 1$, which is enforced by the regulator term, and $\tilde{c}_{n>0}^{\text{ideal}} = 0$. The realistic case, however, is given by

$$\frac{\langle \text{tr} \tilde{W}_{r \times t} \rangle}{\langle \text{tr} \tilde{W}_{r \times 0} \rangle} = \frac{\sum_{n=0}^{\infty} |\tilde{c}_n|^2 e^{-taE_n(r)}}{\sum_{n=0}^{\infty} |\tilde{c}_n|^2}. \quad (33)$$

The ideal case defines our optimum; therefore, the loss function must minimize the difference between the ideal case and the actual case.

The difference between the ideal and the actual case is

$$\Delta_t \equiv e^{-taE_0(r)} - \frac{\sum_{n=0}^{\infty} |\tilde{c}_n|^2 e^{-taE_n(r)}}{\sum_{n=0}^{\infty} |\tilde{c}_n|^2}. \quad (34)$$

From this, we can make some simple assumptions:

$$|\tilde{c}_n| \geq 0, \quad (35)$$

$$E_{n>0}(r) \geq E_0(r) > 0, \quad (36)$$

from which it follows $0 < e^{-ta(E_{n>0}(r)-E_0(r))} < 1$ and thus

$$\Delta_t = e^{-taE_0(r)} \left(1 - \frac{\sum_{n=0}^{\infty} |\tilde{c}_n|^2 e^{-ta(E_{n>0}(r)-E_0(r))}}{\sum_{n=0}^{\infty} |\tilde{c}_n|^2} \right) > 0. \quad (37)$$

Because the difference Δ_t is positive for all t , the ratio $\langle \text{tr} \tilde{W}_{r \times t} \rangle / \langle \text{tr} \tilde{W}_{r \times 0} \rangle$ is smaller than the ideal case. Therefore, maximizing this ratio corresponds to minimizing Δ_t , eventually converging to a correlator dominated by the ground state. Additionally, summing over all t with hyperparameters W_t restores L^{phys} . To recover the full loss function, we introduce L^{reg} for the same reason as in the transfer matrix derivation. We finally obtain the same loss function from two different starting points of our argumentation: one from the Hamiltonian and transfer matrix description and one from the spectral representation of the correlator.

Both chains of arguments leading to the loss function assume a positive transfer matrix. To preserve positivity, we retain the original temporal link variables and remain within the spatial subvolume to prevent any temporal overlap, thereby guaranteeing positivity. If these conditions do not hold, we would have to reformulate the loss function, which is not considered in this study.³ Since the whole argumentation relies only on the structure of a Euclidean time correlator, its principle can be generalized to other states, such as gluelumps, glueballs, or any hadron. Developing the corresponding neural networks is left for future research.

Before concluding this section, we discuss alternative loss functions that lead to the same physics. The first alternative is to minimize or flatten the effective mass directly. Minimizing corresponds to minimizing the energy, and flattening corresponds to reducing excited-state contributions. However, to achieve a good estimate for the effective mass for a single gradient descent step, many field configurations need to be evaluated, typically on the order of 100. This requires computational time, and the signal is lost at larger separations r , which slows down the training process. Equation (29) for defining L^{phys} also works for much smaller batch sizes; in fact, estimating it with a single lattice configuration for each gradient descent step turns out to be enough for computational cost-efficient training.

Another alternative is to minimize the variance of the Wilson loop directly. A robust determination of the variance requires, however, numerous samples, leading

again to a more costly training regimen. Additionally, variance reduction does not guarantee a correlator dominated by the ground state.

IV. APPLICATIONS AND RESULTS

Having introduced the theoretical and technical background in the previous sections, in this section, we proceed to a realistic test problem. In Sec. IV A, we introduce, test, and compare different network architectures on two spatial separations of the Wilson loops. We study which network architecture works well and extend it to train it serially on all separations. After training, we freeze the weights of the optimized network and measure this observable on the full ensemble and, finally, extract the static energies. In Sec. IV B, we will apply the optimized network to measure the static force with an operator consisting of a Wilson loop with a chromoelectric field insertion in one of the temporal Wilson lines, which is a valuable playground to test the networks for additional observables. We follow this with a brief demonstration of combining the neural-network method with the multilevel algorithm in Sec. IV C, which sets the foundations for further precision calculations. In Sec. IV D, we extend the formalism for single ground states to a tower of excited states and let the network find automatically interpolators approximating excited states. In Sec. IV E, we visualize the learned structure of the network by tracing the operations of the network in an explicit algebraic representation of the paths.

In this study, we train with pure-gauge configurations on a $20^3 \times 40$ lattice with the Wilson gauge-field action at $\beta = 6.281$. The training ensemble consists of 6000 configurations used previously [10]. We randomly draw a single configuration set for each optimization iteration. For the final measurement, we generated an additional set of 20 000 lattice configurations, using the overrelaxation and pseudo-heat-bath algorithms in the MILC code [42].

The training and final measurement program was developed in PYTHON using PYTORCH, which uses the automatic differentiation module and the AdamW [43] optimizer. The backward algorithm for the convolutional and partially the exponential activation is explicitly implemented. This saves memory, as the standard implementation requires multiple slicing calls, which do not align with the underlying structure of the layers. Conceptual tests were conducted on a single NVIDIA RTX 4090 GPU, while serial training and evaluation were performed on a cluster with NVIDIA H100 GPUs.

A. Ground-state optimization

There are many possibilities for constructing a network architecture. We vary the number of layers, layer types, and the number of hidden channels to investigate how different choices affect performance. Our approach involves starting with a fixed network topology and then iteratively adding

³In the case of improved actions, we tacitly assume that physical positivity [30] suffices.

TABLE I. Labels for the convolutional-bilinear (ConvBilin) alternating networks with the corresponding number of hidden channels. $N^{(0)}$ is 72 for $r = 3$ and 136 for $r = 7$, and $N^{(n_{\text{final}})} = 1$. The initial network architecture is linear $\rightarrow$ conv $\rightarrow$ bilin $\rightarrow$ conv.

Label	$N^{(1)}$	$N^{(n>1)}$
ConvBilin_1	14	4
ConvBilin_2	14	6
ConvBilin_3	14	8
ConvBilin_4	14	10

new layers, as outlined in Sec. III B, while varying the number of hidden channels. We will compare the best results obtained from networks with different configurations that have a comparable number of hidden channels. This will allow us to select the optimal network based on both training quality and computational cost.

In the first iteration, we study a network architecture of alternating convolutional and bilinear layers. The initial network consists of four layers: linear $\rightarrow$ convolutional $\rightarrow$ bilinear $\rightarrow$ convolutional. (Henceforth, we abbreviate convolutional with “conv” and bilinear with “bilin.”) New “ConvBilin” pairs of layers are added iteratively; the choice of a different number of hidden channels and the corresponding label is listed in Table I.

Figure 2 shows the training history of L^{phys} for $r = 3$ and $r = 7$. We observe a discontinuous change (spike) every time a new ConvBilin pair is added. This behavior is to be expected, as adding a new layer introduces new parameters that push the value of the loss function far away from its current optimal value. Additional training steps are obviously required to thermalize the new setup.

We further see that increasing the number of hidden channels increases the performance of the network; the impact at $r = 7$ is larger as at $r = 3$, indicating that for larger distances a higher number of parameters supports the quality of the signal. Too many parameters, here for $N^{(n>1)}$, lead, however, to instability in the training, which is, again, more likely at $r = 7$ than at $r = 3$. This phenomenon occurs due to increased statistical fluctuations at greater distances, while managing more parameters becomes more challenging. Additionally, we find that higher network complexity does not yield significantly better results beyond a certain point. Apart from the weight decay in the AdamW algorithm, no further regularization procedures, such as clipping of the gradient norm, have been applied, an aspect that should be studied in future research.

Figure 3 shows the training history of a single network for better visibility. We find that both types of insertions, convolutional and bilinear layers, provide a significant boost in training performance and finally reach a saturation point. Furthermore, we cannot observe a qualitative difference between the insertion of a convolutional or bilinear layer. However, in our experience, the bilinear layer causes

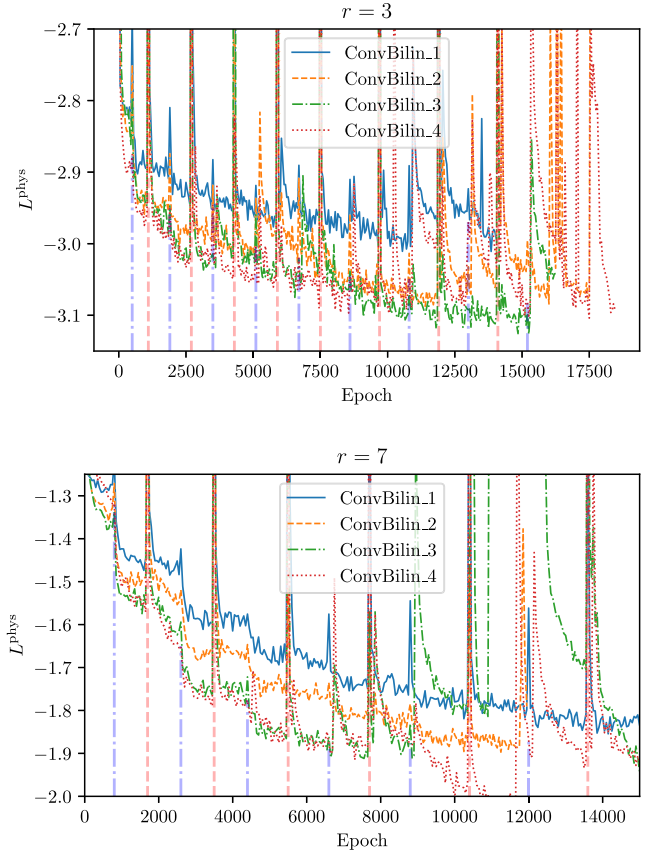


FIG. 2. Comparing the training history for the ConvBilin training procedure. Blue vertical lines indicate the insertion of a convolutional layer, while the red lines indicate the insertion of a bilinear layer. The x axis represents the epoch, i.e., the training time, while the y axis displays the value for L^{phys} only (not the full loss function). A lower value for L^{phys} corresponds to a more optimized result.

numerical instabilities if not treated with great care. Therefore, we will focus on networks without bilinear layers for the rest of this study.

We introduce two additional network concepts: the “ConvExp” and the “ConvLimNeighbor.” The ConvExp network consists solely of convolutional layers, aside from the initial linear layer, with each layer followed by an exponential activation function. This design leverages the regularization properties and multilinear structure of the exponential action.

In contrast, the ConvLimNeighbor network adopts a different philosophy. Instead of adding new convolutional layers with a fixed number of hidden channels, it incorporates a convolutional layer with fewer hidden channels that operates solely over adjacent sites. The new objects are integrated with the preceding layer by combining it with a linear layer that restores the standard number of hidden channels. The initial weights of this linear layer are chosen such that the new construction acts like an identity. This approach also allows the new weights to be randomly

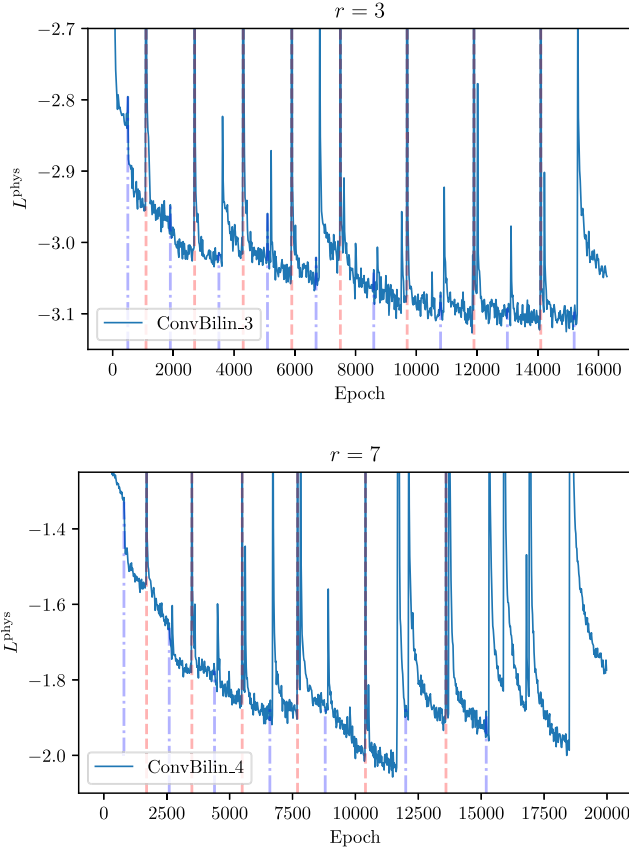


FIG. 3. The training histories for each r of a single network. Blue vertical lines indicate the insertion of a convolutional layer, while the red lines indicate the insertion of a bilinear layer. The x axis represents the epoch, i.e., the training time, while the y axis displays the value for L^{phys} only (not the full loss function). A lower value for L^{phys} corresponds to a more optimized result.

initialized, while the weights of the additional linear layer are set to ensure that the new convolutional layer has a neutral effect. This avoids the potential issue of a symmetric gradient for the new weights, which, although rare, could be problematic.

Moreover, the network's capacity can be expanded by adding more neighboring convolutional layers that require fewer weights than those typically expected with the standard number of hidden layers. We anticipate greater statistical fluctuations as the network becomes thicker, and we do not expect to achieve high information density from distant convoluted lattice sites. This design provides additional stability during training for the ConvLimNeighbor network architecture.

Figure 4 shows the training history of the new network concepts and compares them with the previous ConvBilin networks. We observe that the ConvExp network behaves similarly to the ConvBilin networks; however, it performs better at $r = 7$, although it remains challenging to control. In contrast, the ConvLimNeighbor network demonstrates reliable stability over a wide range of training steps.

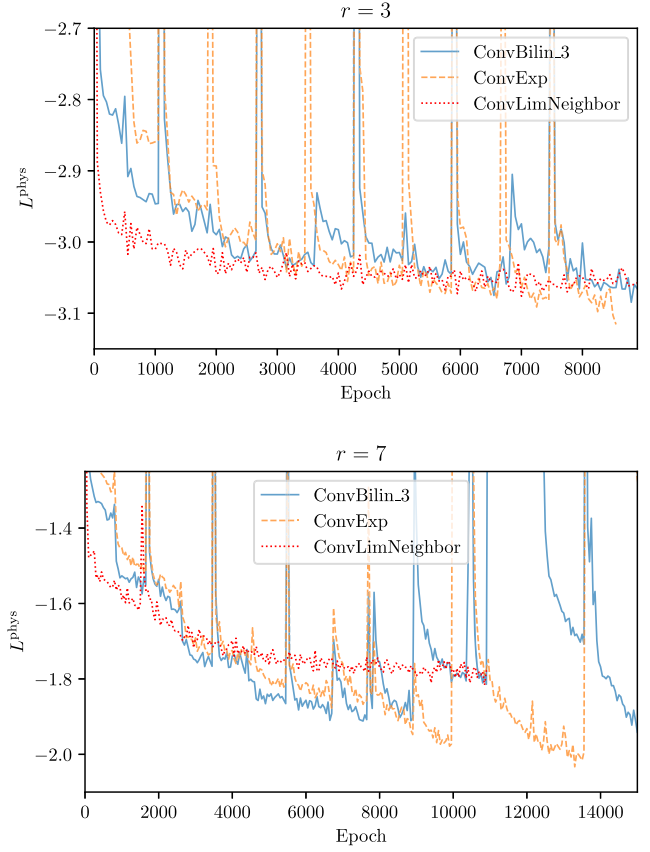


FIG. 4. The training histories for each r of three different network architecture concepts. We obtain similar peak structures for the ConvExp as for the previous ConvBilin network. In contrast, the ConvLimNeighbor exhibits stability over a wide range in the training history.

Nevertheless, for $r = 7$, the ConvLimNeighbor network does not achieve optimal performance in the loss function compared to the other networks. Nonetheless, stability is a crucial property for serially training networks across various r values without the need for fine-tuning each individual case.

To cover all values of r , we choose the ConvLimNeighbor network architecture and train a separate network for each value of r ranging from 1 to 10. As this network architecture is stable, we increase the number of hidden channels as r increases, assuming that this leads to better performance for larger values of r . We add new layers iteratively until the training becomes unstable or the program runs out of memory, at which point it fails automatically. Each network is saved before a new layer is added, allowing us to select a set of networks for final evaluation and comparison. For this computation, we freeze the weights and treat the network as an observable, measuring it across the full ensemble.

Figure 5 presents the bare, normalized correlator from the final measurement for $r = 4$ and $r = 7$. We compare the results obtained from the neural network with those from

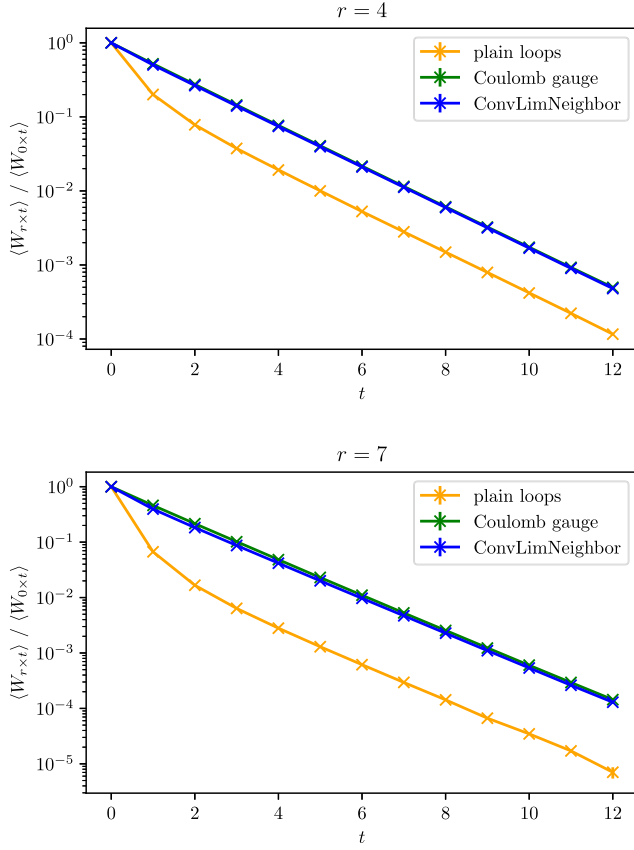


FIG. 5. The bare, normalized correlator from the final measurement for $r = 4$ and $r = 7$ in a logarithmic plot. The data points are connected with lines to guide the eye.

Coulomb-gauge-fixed Wilson-line correlators and plain Wilson loops. While the plain Wilson loops are affected by excited-state contamination at small t , the network correlators and Coulomb-gauge Wilson lines appear nearly identical in a logarithmic plot, suggesting a dominant single exponential decay behavior. For larger t , all three correlators exhibit the same decay rate, indicating that they represent the same ground state. Notably, especially in the $r = 7$ plot, the signal for the plain Wilson loops deteriorates more quickly than that of the neural-network Wilson loops and Coulomb-gauge Wilson lines. In summary, our observations suggest that both the neural network and the Coulomb-gauge Wilson-line correlator yield comparable results.

Next, we analyze the effective masses of the correlators defined by the following equation:

$$am_{\text{eff}}(t) = -\ln \frac{\langle \text{tr } W_{r \times (t+1)} \rangle}{\langle \text{tr } W_{r \times t} \rangle}, \quad (38)$$

where W represents either the neural-network Wilson loop, the plain Wilson loop, or the Coulomb-gauge Wilson lines. Figure 6 illustrates the effective mass plots for the three correlators. Both the neural-network Wilson loop and the

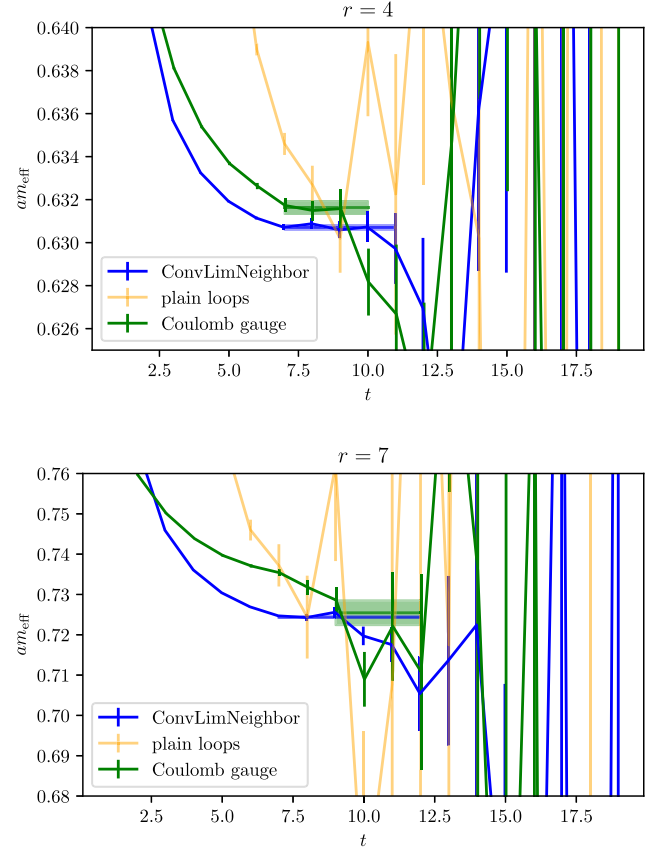


FIG. 6. The effective masses for the different correlators at $r = 4$ and $r = 7$.

Coulomb-gauge Wilson lines demonstrate a significantly better signal and much less excited-state contamination than the plain Wilson loops. However, at small t (up to five or six), the Coulomb-gauge Wilson lines and the neural network still suffer from some excited-state contamination. Furthermore, the neural network performs comparably well or even better than the Coulomb-gauge Wilson lines in terms of signal-to-noise ratio and the range of the plateau. Therefore, we conclude that the neural-network approach is as effective as the Coulomb-gauge method while still providing gauge-invariant operators. Reducing further the excited-state contamination at small t remains an open challenge for future studies.

A relevant question is how different network architectures influence their final physical outcomes. So far, we have introduced three different network architectures. Figure 7 shows the effective mass for all three. We notice that all three networks show the same plateau range and value. The primary distinction lies in the excited-state contamination at small t , which we aim to minimize. The difference between the ConvBilin network and the other two architectures disappears when a more complex ConvBilin network is used; this configuration was selected for demonstration purposes. Since the final physical outcome depends only slightly on the network architecture,

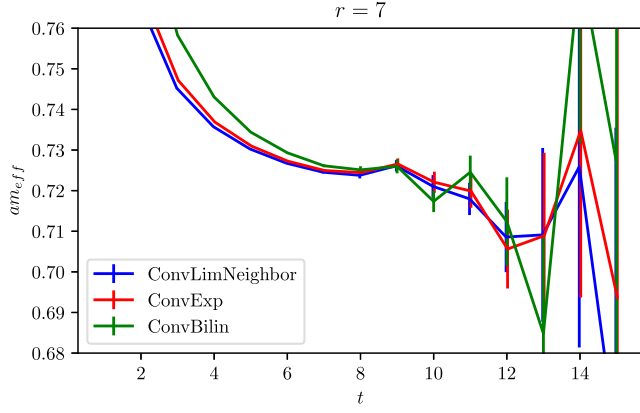


FIG. 7. The effective mass for different network architectures.

the optimal network is assessed based on signal quality and computational cost. More complex networks typically require greater computational resources. However, in certain situations, prioritizing networks that minimize excited-state contamination—regardless of computational cost—can be advantageous.

To extract the static energy of the ground state

$$aE_0(r) = \lim_{t \rightarrow \infty} am_{\text{eff}}(t)|_r, \quad (39)$$

we perform a plateau fit by selecting a reasonable time range and propagating the statistical error through blocking and jackknife resampling. An example of this analysis is shown in Fig. 6, showing statistical uncertainties only.

Figure 8 presents a comparison of the static energy obtained from the neural-network approach and the Coulomb-gauge method. Both methods yield similar results, although discrepancies arise at larger values of r . These discrepancies can be attributed to the typical systematic differences from excited-state contamination that arise when comparing different correlators.

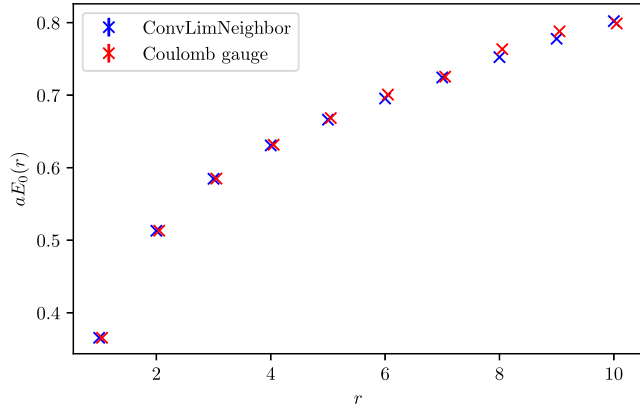
FIG. 8. The static energy as a function of the distance r for the neural-network approach in comparison to the Coulomb-gauge approach. Statistical error bars are included but not visible.

TABLE II. Comparison of the final, measured physical loss function with the theoretical optimal value obtained from the mean values of the effective masses.

r	$-L^{\text{MPhys}}$	$-L^{\text{OPhys}}$	$-\Delta L$	$\frac{L^{\text{MPhys}}}{L^{\text{OPhys}}}$
1	5.171	5.205	0.034	0.993
2	3.636	3.712	0.076	0.979
3	3.081	3.210	0.130	0.960
4	2.758	2.939	0.181	0.939
5	2.478	2.751	0.273	0.901
6	2.193	2.611	0.419	0.840
7	1.974	2.481	0.507	0.796
8	1.715	2.363	0.648	0.726
9	1.438	2.266	0.828	0.634
10	1.147	2.177	1.030	0.527

We conclude that the neural-network method introduced in this study is as effective as the Coulomb-gauge method. This is significant because the neural network provides a gauge-invariant operator that can be used for further processing and can be applied in situations where implementing the Coulomb gauge is not straightforward. Furthermore, the neural-network approach can be integrated with other signal-improving techniques, such as hypercubic (HYP) smearing [44], gradient flow, and multilevel algorithms. A different application of the optimized operator is to apply it directly to ground-state expectation values or as an input state for a correlation matrix measurement. We will explore those possibilities in the following sections.

The static energies extracted from the plateau fit define the ideal value of the loss function. So far, we have optimized the loss function without knowing its theoretically optimal value. This situation corresponds to a single exponential of the correlator, where the mass precisely matches the static energy. Therefore, by inserting this exponential function with the static energy in the exponent into the loss function, we represent the ideal case, which we then compare to the value obtained from the measured correlator.

Table II compares the measured loss function L^{MPhys} with the optimal value L^{OPhys} . We take differences $\Delta L = L^{\text{OPhys}} - L^{\text{MPhys}}$ and ratios $L^{\text{OPhys}}/L^{\text{MPhys}}$ of both values. The loss function serves as an absolute measure, where a value of 0 indicates a correlator without any correlation. Its minimum value is defined by the static energy exponential. Thus, the ratio of the measured to the ideal value reflects the relative success of the neural network.

We observe that for $r = 1$ the success rate is nearly 1, gradually decreasing to around 0.5 as r increases. This trend suggests that more complex networks may be necessary for larger distances, as one can expect more intricate gluon dynamics in such cases. Further improvements to the networks will be explored in future studies.

B. Static force as a ground-state expectation value

The static force is defined as the derivative of the static energy:

$$F_{\partial_r}^{\text{cont}}(r) = \frac{\partial E_0(r)}{\partial r}. \quad (40)$$

This quantity is physical and therefore ultraviolet and infrared finite. On the lattice, the static force is approximated using a numerical derivative of the static energy. For example, by applying a symmetric two-point derivative, we can express it on the lattice as

$$a^2 F_{\partial_r}(r) = \frac{1}{2} [aE_0(r+1) - aE_0(r-1)]. \quad (41)$$

In this equation, $aE_0(r)$ represents the static energy measured from the $t \rightarrow \infty$ limit of the effective masses of the Wilson loops.

The static force is used for scale setting through the indirect equation

$$r^2 F_{\partial_r}(r)|_{r=r(c)} = c. \quad (42)$$

A common choice is the Sommer scale [45] defined as $r_0 \equiv r(1.65)$, which is widely used to set the lattice spacing. Other common choices are $r_1 \equiv r(1)$ [46] and $r_c \equiv r(0.65)$.

This definition is accompanied by systematic errors stemming from the numerical derivative of the static energy. Therefore, an alternative definition was introduced in [8,15,47] as

$$\begin{aligned} F_E(r) &= -\lim_{t \rightarrow \infty} i \frac{\langle \text{tr} \{ W_{r \times t} \hat{\mathbf{r}} \cdot g \mathbf{E}(\mathbf{r}, t^*) \} \rangle}{\langle \text{tr} W_{r \times t} \rangle} \\ &= -\lim_{t \rightarrow \infty} i \frac{\langle \text{tr} W E_{r \times t} \rangle}{\langle \text{tr} W_{r \times t} \rangle}, \end{aligned} \quad (43)$$

where $\hat{\mathbf{r}}$ is the direction vector of the quark-antiquark separation, $\mathbf{E}$ is the chromoelectric field vector, t^* is an arbitrary time slice, and g is the bare coupling. The object in the numerator, with the shorthand notation $W E_{r \times t}$, is a Wilson loop of spatial extent $\mathbf{r}$ with a chromoelectric field insertion at t^* in the Wilson line at $\mathbf{r}$. The notation implies, but does not spell out, the path ordering of the modified Wilson loop.

The $t \rightarrow \infty$ limit yields ground-state dominance, so $F_E(r)$ converges to the ground-state expectation value of the chromoelectric field [9]. Therefore, this object can be interpreted as

$$F_E(r) = \frac{\langle \Psi_0 | -i \hat{\mathbf{r}} \cdot \mathbf{E}(\mathbf{r}) | \Psi_0 \rangle}{\langle \Psi_0 | \Psi_0 \rangle}, \quad (44)$$

where $|\Psi_0\rangle$ is the ground state of the static quark-antiquark pair separated by $\mathbf{r}$. In previous studies, e.g., Refs. [9,10], only the plain Wilson loop was considered as the trial state,

but we expect a better overlap could arise from optimizing a generalized Wilson loop. Thus, we investigate here how we can use the neural-network methods developed above to improve the overlap with the ground state, thereby enhancing the overall measurement.

On the lattice, we discretize the $\mathbf{E}$ field, and multiple choices exist. We pick the clover leaf discretization; for details, see Ref. [10]. These discretizations require renormalization, introduced as

$$Z_E F_E(r) = F_{\partial_r}(r). \quad (45)$$

In principle, Z_E is independent of r , while practical implementations tend to show a mild, residual dependence. A nonperturbative definition of Z_E is thus

$$Z_E = \frac{F_{\partial_r}}{F_E}. \quad (46)$$

In this study, we will focus only on the comparison between measuring the static force with and without the neural-network-optimized Wilson loops. A previous study used the gradient flow method to renormalize the $\mathbf{E}$ field [10].

We measure the $\mathbf{E}$ -field insertion in the ConvLimNeighbor-optimized Wilson loop and compare it with the $\mathbf{E}$ -field insertion in the plain Wilson loop. Figure 9 shows a comparison between both measurements. We observe a significant improvement in the signal with the neural-network Wilson loop while both measurements approach the same value.

To extract the $t \rightarrow \infty$ limit, we carry out a plateau fit akin to the effective mass extraction used for determining the static energy. We select an appropriate range in t and perform constant fits for the neural-network Wilson loops. However, the signal from the plain Wilson loops does not provide sufficient data for a reliable plateau extraction.

Figure 10 shows the static force as a function of r , obtained via both Eqs. (41) and (43). In both cases, the neural-network Wilson loop is shown. We observe a multiplicative constant shift between both curves, which is expected due to the required renormalization Z_E of the $\mathbf{E}$ field introduced in Eq. (45).

We have demonstrated that neural-network Wilson loops can significantly enhance the measurement of ground-state expectation values for the chromoelectric field. Because the improvement stems from the faster approach to the ground state, we expect the enhancement will hold for various other operators, too. Additionally, this method is especially beneficial in situations—as here with the static-force matrix element—where applying the Coulomb gauge is not straightforward. Overall, improving the measurement of matrix elements is crucial for various applications in lattice QCD, spanning from static force calculations to hadronic expectation values.

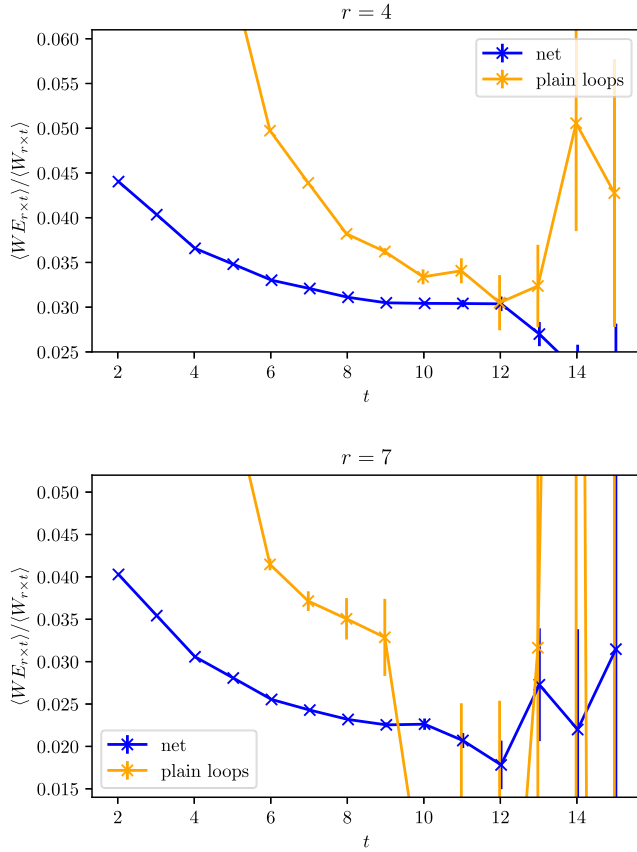


FIG. 9. The t dependence of the E -field insertion in the Wilson loop for two different distances r . The figures show the comparison of the ConvLimNeighbor-optimized Wilson loop with the plain Wilson loop. We observe that the quantities approach a plateau, where the neural-network Wilson loop exhibits a significantly improved signal and a more robust plateau. The neural network is clearly superior to the plain Wilson loop.

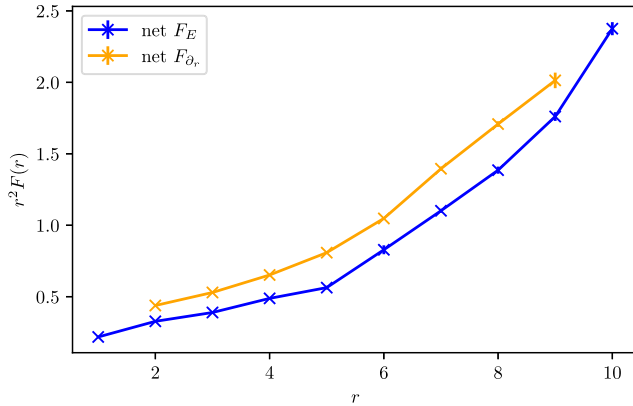


FIG. 10. Comparison of the static force obtained through the numerical derivative, Eq. (41), and the chromoelectric-field insertion, Eq. (43). We notice a multiplicative constant shift between both results, which is attributed to the renormalization Z_E of the E field introduced in Eq. (45).

C. Neural networks with the multilevel algorithm

In this section, we provide a brief overview of how the neural-network method can be integrated with existing algorithms and methods. In particular, we demonstrate how the multilevel algorithm [48] can be used to further improve the signal. The multilevel algorithm applies here because we only study the pure gauge theory using the ultralocal Wilson action. The multilevel algorithm can be applied to the temporal links because the neural network affects only the spatial part.

After optimizing the Wilson loop with a neural network, the largest source of statistical fluctuation is found to originate from the temporal Wilson lines. In the multilevel algorithm, we sample the temporal-link variables N_{ml} additional times. Since the Wilson action is local, we can treat each time slice independently from the other time slices. The final Wilson loop is constructed by combining all possible temporal slices, which increases the effective sample size by a factor of order N_{ml}^t for Wilson loops of temporal extent t .

The multilevel object is computed from multilevel temporal link variables

$$\mathbb{U}_{ABCD}(\mathbf{x}, \mathbf{r}, t) \equiv \frac{1}{N_{\text{ml}}} \sum_{i=1}^{N_{\text{ml}}} U_4^{(i)}(\mathbf{x} + \mathbf{r}, t)_{AB} U_4^{(i)\dagger}(\mathbf{x}, t)_{CD}, \quad (47)$$

where i sums over the different additionally sampled time slices and the indices A, B, C , and D are the indices for the components of the 3×3 matrices. The multilevel temporal-link variables form the multilevel temporal Wilson line as

$$\mathbb{T}_{ABCD}(\mathbf{x}, \mathbf{r}, t) = \mathbb{T}_{AEFB}(\mathbf{x}, \mathbf{r}, t-1) \mathbb{U}_{ECDF}(\mathbf{x}, \mathbf{r}, t), \quad (48)$$

where the Einstein summation rule applies, and at the start

$$\mathbb{T}_{ABCD}(\mathbf{x}, \mathbf{r}, t)|_{t=1} = \mathbb{U}_{ABCD}(\mathbf{x}, \mathbf{r}, 0). \quad (49)$$

The multilevel-improved Wilson line is computed as

$$\text{tr} \tilde{W}_{r \times t}|_{\text{ml}} = \tilde{S}(\mathbf{x}, \mathbf{x} + \mathbf{r}, 0)_{AB} \times \mathbb{T}_{BCDA}(\mathbf{x}, \mathbf{r}, t) \tilde{S}^\dagger(\mathbf{x}, \mathbf{x} + \mathbf{r}, t)_{CD}, \quad (50)$$

where $\tilde{S}$ can also be replaced by the straight line S .

We use the 6000 lattice configuration employed for training as initial configurations to sample additional $N_{\text{ml}} = 20$ temporal-link variables. Since the spatial part with the neural network is composed only of spatial link variables, we need to compute it only for the initial configurations and keep it fixed for the contraction of the multilevel temporal Wilson lines. Thus, this method is an efficient approach to improve the signal.

Figure 11 shows two examples of effective masses for the multilevel-improved Wilson loops. We observe a significant improvement for both the plain Wilson loops

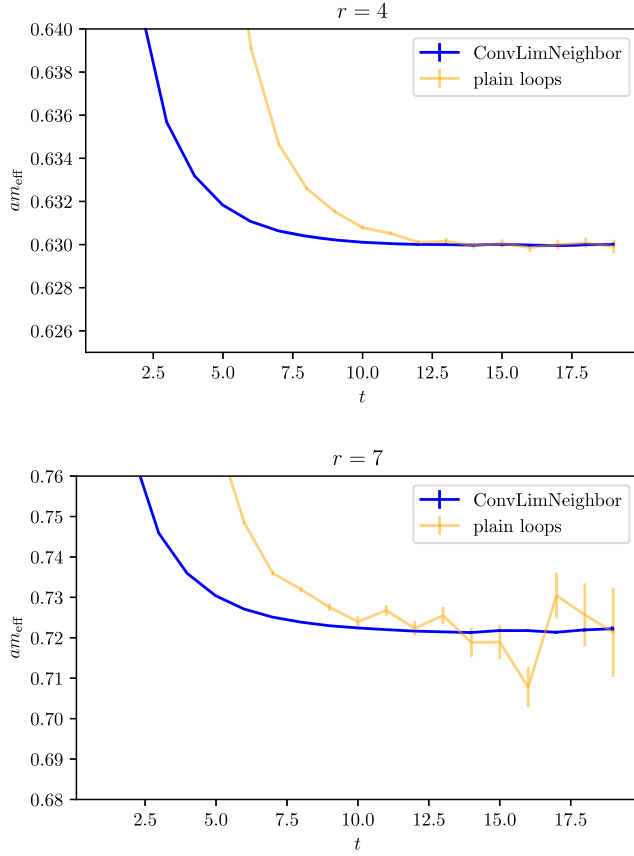


FIG. 11. Two examples of the effective masses for the multilevel-improved Wilson loops. It shows the multilevel-improved Wilson loops for the plain Wilson loop and the neural-network Wilson loop.

and the neural-network Wilson loops. At larger distances, here $r = 7$, the neural-network Wilson loop preserves a stable signal while the plain Wilson loop is already losing the signal. This example further demonstrates the utility of neural networks for Wilson loops.

D. Seeking excited states

In the previous sections, we studied the ground state of the static quark-antiquark pair and optimized the interpolator to minimize excited-state contamination. The excited states themselves are particularly important for understanding gluon dynamics and the spectrum of exotic states, and they are referred to as hybrid static energies because they model hadrons with a valence gluon and a heavy $Q\bar{Q}$ pair. They are characterized by a specific set of quantum numbers, related to the symmetry of the two static sources at a fixed distance in QCD. To obtain hybrid energy levels, the static energies are input for the Born-Oppenheimer effective field theory for hybrids [6,17]. They have been calculated on the lattice since the seminal work of Juge *et al.* [32–34].

Before showing results from our neural networks, let us review the symmetry properties of the spectrum. On the

lattice, these symmetries include four discrete rotations around the separation axis, which correspond to the cubic rotation group: $R_{\pi/2}$, R_{π} , $R_{3\pi/2}$, and $R_{2\pi} = 1$. Additionally, there are reflections at a plane spanned by the separation axis and an orthogonal direction (for example, $\mathcal{P}_x$ or $\mathcal{P}_y$ if z is the separation axis). There is also a combined operation of charge conjugation followed by a reflection at the center point of the separation axis, $(\mathcal{P} \circ \mathcal{C})$.

The cubic rotations generate the cubic rotation group, whose irreducible representations are related to continuum angular momentum. For the three lowest angular momentum states, the labels $\Lambda = \Sigma, \Pi, \Delta$ suffice, where Σ corresponds to the $J = 0$ state, Π to the $J = 1$ state, and Δ to the $J = 2$ state.

The eigenvalues of the reflections $\mathcal{P}_{x/y}$ are labeled by $\epsilon = \pm$. The eigenvalues of the combined operation $\mathcal{P} \circ \mathcal{C}$ are denoted as $\eta = g$ (for even) and u (for odd). Thus, the complete notation for a state is given by $\Lambda_{\eta}^{\epsilon}$. For example, the static quark-antiquark ground state is denoted Σ_g^+ .

It is important to note that for $J \geq 1$, the two states with $\epsilon = \pm$ are degenerate. Thus, the states Π_{η} and Δ_{η} form doublets. To obtain the excited states, interpolators of superpositions of differently shaped paths, obeying the correct quantum numbers, can be constructed; recent extensive studies can be found in [18,20].

The Π_u state is the first excited state and therefore of particular interest. It has $J = 1$ angular momentum, which is generated by the $\pi/2$ rotations, each assigned with a phase shift of $e^{i\pi/2} = i$. Hence, given a trial path S_{trial} , the $J = 1$ state is constructed by $(1 + iR_{\pi/2} - R_{\pi} - iR_{3\pi/2})S_{\text{trial}}$. An example for a rectangular trial path is shown in Fig. 12. Furthermore, a reflection at the center point yields an overall minus sign, defining the odd property under the $\mathcal{P} \circ \mathcal{C}$ transformation for the Π_u state. This pictorial understanding is crucial to understanding Sec. IV E, where we unveil the neural network to see what it has learned.

In general, a defining property of excited states is their orthogonality. Given a tower of states of a theory $|\psi_m\rangle$ numerated by m (m can also be a multi-index, like for the hybrid states), the states fulfill $\langle \psi_{m'} | \psi_m \rangle \propto \delta_{m'm}$. The states

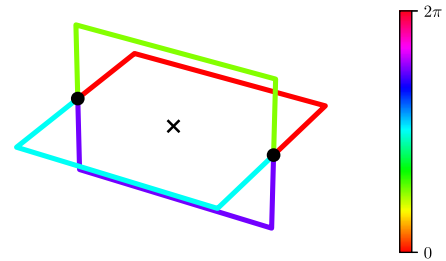


FIG. 12. Illustrative picture for a Π_u state. The black dots note the static quarks, and the cross marks the center point of the separation. The paths are color coded according to their complex phase.

are ordered such that $E_{m'} \geq E_m$ if $m' > m$, where $m = 0$ defines the ground state and the remaining states are the excited states. If $E_{m'} = E_m$ even if $m' \neq m$, the states for m' and m are degenerate, as in the doublets for Π_η and Δ_η discussed above.

The orthogonality leads to two major conclusions. First, the spectral representation Eq. (11) only sums over states with the same quantum number of the initial state, i.e., $c_n = 0$ if $n \neq m$. Second, the final, optimized state from Eq. (27) has the same quantum number as the trial states $\tilde{S}$. In conclusion, we can apply Eqs. (29) and (30) on a neural network that represents a state with the desired quantum number. Therefore, we can use the established procedure to optimize for the “ground state” within the domain of a fixed quantum number.

To obtain neural-network trial states of excited states, we have two options. For the first option, we construct neural networks that explicitly obey a specific quantum number. This method establishes constraints on the network weights and, hence, requires more technical developments. In the second option, we create a set of trial states without any constraints, orthogonalize the set, and apply the loss function for each orthogonalized state individually. This method requires only a minor technical modification, but we cannot determine the explicit quantum numbers without additional measurements. In this work, we will focus on the second option, whereas the first holds promising prospects for future studies.

We modify the neural network by changing the number of final output elements, i.e., $N^{(n_{\text{final}})} > 1$. The states are therefore labeled $\tilde{S}_i(\mathbf{x}, t) = \phi_i^{(n_{\text{final}})}(x)$, $i = 1, \dots, N^{(n_{\text{final}})}$. Here, we focus specifically on $N^{(n_{\text{final}})} = 8$. The neural-network Wilson loop composed of $\tilde{S}_i(\mathbf{x}, 0)$, $\tilde{S}_j(\mathbf{x}, t)^\dagger$, and the two temporal Wilson lines then yields a correlation matrix as $C_{ij}(t) = \langle \text{tr} \tilde{W}_{ij,r \times t} \rangle$. We employ the generalized eigenvalue problem (GEVP)

$$C(t)v_n(t, t_0) = \lambda_n(t, t_0)C(t_0)v_n(t, t_0) \quad (51)$$

to orthogonalize the set of states [49–51], where n labels the tower of orthogonal (excited) states, λ_n is the generalized eigenvalue, i.e., the correlation function of the corresponding (excited) state with $\lambda_n(t_0, t_0) = 1$, $v_n(t, t_0)$ is the generalized eigenvector, and t_0 is a reference time. We focus here on $t_0 = 0$; in general, the dependence of the final result on t_0 needs to be considered.

The new loss function is defined as

$$L = L^{\text{phys,GEVP}} + L^{\text{reg}} + L^{\text{ortho}}, \quad (52)$$

where the new individual components are given by

$$L^{\text{phys,GEVP}} = \sum_i^{n_{\text{final}}} W_i^s \left(- \sum_{t=1}^{t_{\text{max}}} W_t \lambda_i(t, t_0) \right), \quad (53)$$

which is a generalization of Eq. (29) to all excited states with an additional set of hyperparameters W_i^s , and

$$L^{\text{ortho}} = \sum_{i>j} \left(\frac{\langle \text{tr} \tilde{W}_{ij,r \times 0} \rangle}{\sqrt{|\langle \text{tr} \tilde{W}_{ii,r \times 0} \rangle| |\langle \text{tr} \tilde{W}_{jj,r \times 0} \rangle|}} \right)^2, \quad (54)$$

which enforces orthogonality of the neural network. The term L^{ortho} is not necessary for the physics but can help to maximize the learned information in the initial set of states.

We employ the previously introduced ConvLimNeighbor architecture, as it has demonstrated high stability in the training history. The number of final outputs is $N^{(n_{\text{final}})} = 8$, and the final layer is a linear layer with complex weights. As above, we use progressive learning. Approximating the correlation matrix with a single gauge configuration is sufficient. After training, we select the best networks, freeze the weights, and perform the final measurement of the correlation matrix over the whole ensemble.

Figure 13 shows the final correlator of the excited states, i.e., the functional form of the generalized eigenvalues in a logarithmic scale for two separations r . We observe that the ground state ($n = 0$) exhibits the same mass as the state from the single-state network. This is expected because all approaches should converge to the same ground state. Furthermore, we see that the first two excited states ($n = 1, 2$) share the same mass. This is also to be expected since the next-highest states are in the doublet Π_u . The third state ($n = 3$) has a different mass than the states below and above, which is again anticipated, since this state should be the singlet Σ_u^- . Overall, we conclude that this method effectively generates interpolators for a set of optimized excited states in an automated manner. This is a significant achievement, considering that constructing a set of interpolators is typically a challenging task.

Figure 14 shows the effective masses of the first three states. The first (ground state) exhibits a clear plateau; the first two excited states converge to a common plateau, although the signal is lost for larger times. We extract the plateaus by constant fits in a reasonable fit range.

Figure 15 shows the r dependence of the first three states' static energies. We see that the static energies of the first two excited states do not exhibit a Coulomb-like ($\propto 1/r$) behavior. Instead, they follow a slightly repulsive behavior. For larger r , the r dependence is milder. This shape of the functional curve is exactly what we would expect for the static energies of Π_u ; see, for example, Fig. 2 of Ref. [20]. It is the octet repulsion predicted for the hybrid static energies in Born-Oppenheimer effective field theory [4,17].

The evidence presented in Figs. 13–15 strongly suggests that the first excited state corresponds to the Π_u state, while the ground state corresponds to the Σ_g^+ state. Because we

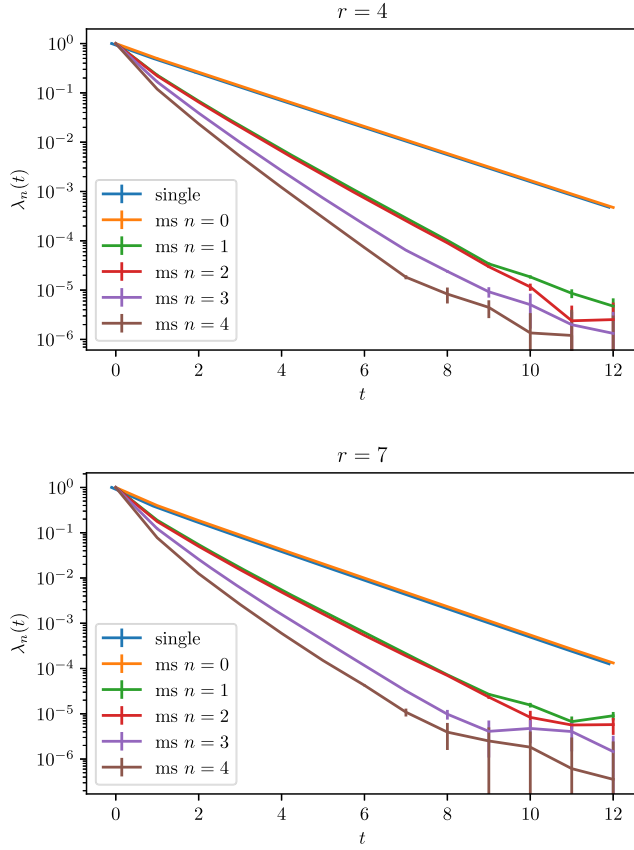


FIG. 13. The generalized eigenvalue from the final measurement for $r = 4$ and $r = 7$ in a logarithmic plot. The data points are connected with lines for better visualization. The label “single” indicates the measurement for the single state network, while “ms $n = _$ ” indicates the actual eigenvalues from the final measurement. As expected, the ground state of the tower of states has the same mass as the single state network.

have not specified the quantum numbers explicitly, it is possible that the network could have identified an unanticipated state between Σ_g^+ and Π_u . This possibility could be more definitively investigated by searching for excited states at off-axis separations and cross-checking the energy levels from all orientations, thereby assembling the Π_u doublet and, perhaps, other multiplets. Constructing off-axis interpolators will be a challenge, as the cubic group theory for off-axis separations is more complicated and has not yet been addressed.

In conclusion, we have demonstrated that the neural-network approach can effectively identify excited states without requiring additional constraints. This discovery opens up new possibilities for future applications, especially in cases where the system at hand is not yet fully understood.

E. Unveiling the networks

The final network consists of a superposition of differently shaped paths connecting the starting and end points,

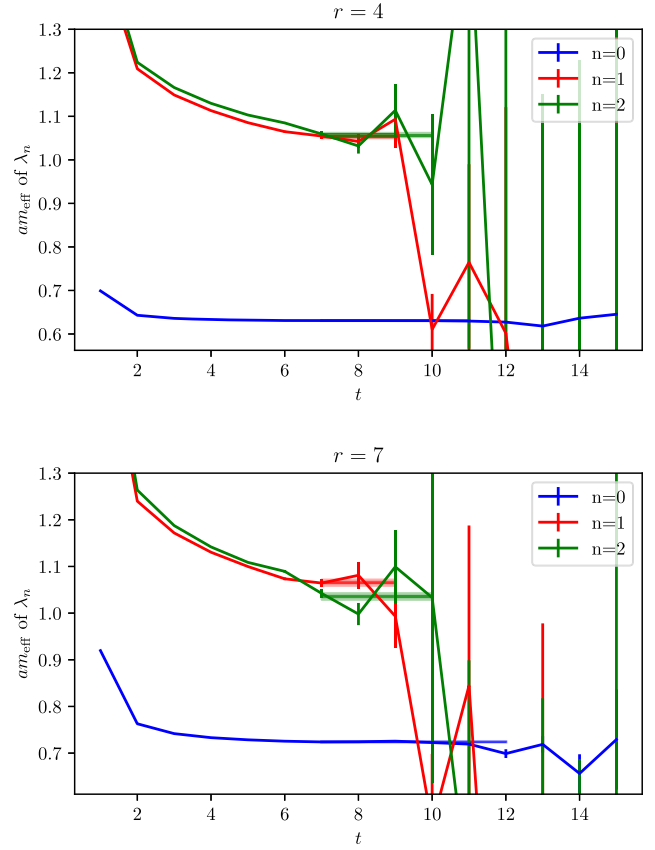


FIG. 14. The effective masses of the first three states for $r = 4$ and $r = 7$.

with each path weighted by an optimized coefficient. The actions of the layers and, hence, of the network are clearly defined and can be used to algebraically trace the learned structure of the neural network. In this section, we introduce a formalism to describe and then *visualize* the neural network’s superposition of three-dimensional paths.

The path of a straight Wilson line, along with all possible plaquette insertions, can be explicitly represented in

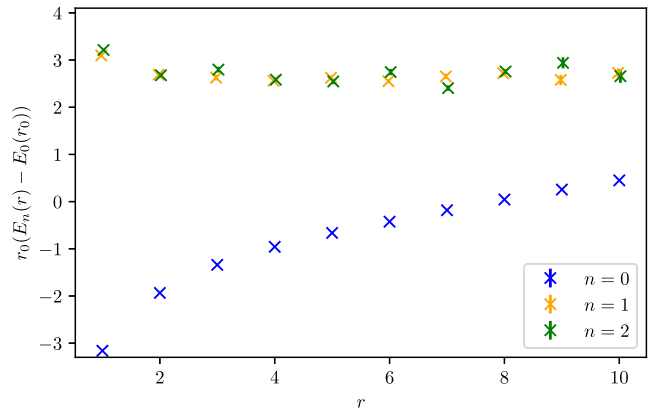


FIG. 15. The static energies of the first three states as a function of the distance r .

algebraic form. For instance, if $r = 3$ and $\hat{\mathbf{e}}_r = \hat{\mathbf{e}}_x$, the straight line might be represented as (x, x, x) , where each element in the sequence corresponds to a step in the directions $\pm x$, $\pm y$, or $\pm z$. Likewise, an example for a plaquette-inserted Wilson line is $(y, x, -y, x, x)$ which corresponds to the plaquette insertion $\phi_{0,y,x}$ as defined in Eq. (17). By following this method, we can derive an algebraic representation for each initial element.

A sum of paths, as created in the linear layer, can be represented as a sequence of those paths, each with an assigned weight. A sequence of paths p_i with weights w_i can be written as $[(w_1, p_1), (w_2, p_2), \dots]$. Adding a second superposition of the same paths, now with weights w'_i , is given by $[(w_1, p_1), \dots] + [(w'_1, p_1), \dots] = [(w_1 + w'_1, p_1), \dots]$. This prescription enables us to analytically trace the action of the linear layer by following the definition in Eq. (19), and, thus, it can be used to extract the learned structure of the neural network.

The convolutional layer modifies the path directly. For instance, the introduced straight line and the plaquette insertion changes by a convolution in z direction into $(z, x, x, x, -z)$ and $(z, y, x, -y, x, x, -z)$, respectively. The new paths are assigned a weight and summed up with a set of other convoluted paths. Therefore, the actions of the convolutional layer, as defined in Eq. (21), can also be algebraically traced and used to extract the learned structure of the neural network.

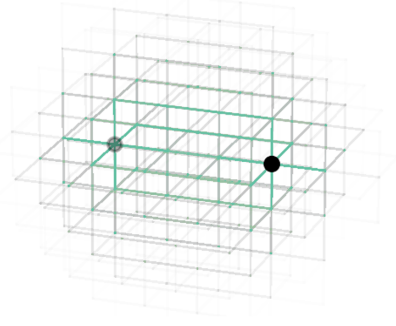
We can establish similar rules for the bilinear layer. Here, however, we focus exclusively on convolutional and linear layer networks for reasons mentioned below.

Overall, the neural network effectively maps the set of straight Wilson lines and their plaquette-inserted modifications into a sum of less-trivially shaped Wilson lines represented as an algebraic sequence of weights and the corresponding algebraic form of the path. These weights are dictated by the neural network's parameters, and we obtain the optimized set after training the network.

We focus on a neural-network architecture consisting only of linear and a few convolutional layers. With bilinear layers and numerous convolutional layers, we found that storing all weights and their corresponding paths becomes memory demanding, particularly for networks that contain bilinear layers. Therefore, for visualization we use neural networks with one linear and three convolutional layers each. These networks are less optimal in terms of ground-state overlap than the more complex ones used in the rest of this study. They should be good enough to provide a first impression of how the neural network learns an optimal superposition of shapes.

By tracing the analytical operations of the layers using the optimized values of the neural network's weights, we compute the weight of each path that appears in the optimized operator for the static quark-antiquark ground state. We then visualize these paths in a three-dimensional

$$r = 3$$



$$r = 7$$

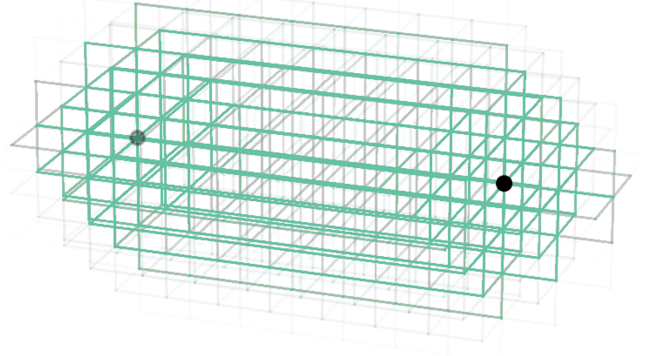


FIG. 16. Three-dimensional visualization of the neural-network-optimized operator for the ground state of a quark-antiquark pair at particle separations $r = 3$ and $r = 7$. The black dots mark the positions of the sink and source of the static pair, and the turquoise lines represent the Wilson lines that are present in the optimized operator. The opacity of each line is proportional to the absolute value of the weight in the optimized operator.

plot, with the opacity of each path being proportional to the absolute value of its weights.

The resulting picture is shown in Fig. 16 for two quark-antiquark separations, $r = 3$ and $r = 7$. In both cases, the neural-network-optimized Wilson line is several lattice spacings thick. The contribution of link variables for paths that are farther away from the central straight line decreases gradually, indicating a radial decay of the gluon dynamics. The learned structure resembles a flux tube, where gluonic energy is concentrated in a cylinder around the separation axis. Additionally, the flux tube appears to have a rotational symmetry around the separation axis and a reflection symmetry around the center of the separation. This is expected since the ground state Σ_g^+ is defined by those quantum numbers.

Now we repeat the procedure for the excited state. At this stage, we focus on a neural network with one linear layer and one convolutional layer, as deeper networks were too memory demanding. A more efficient algorithm is to be developed in future studies.

In contrast to the single-state network, the excited-state neural network has multiple outputs, where the correct sum of those outputs generates the corresponding state. The correct weights for each output are not initially known, and a measurement over the lattice ensemble is required to perform the GEVP. We perform the measurement of the correlation matrix and use the generalized eigenvalue $v_n(t, t_0)$ as the coefficient vector for the final sum.

Nevertheless, a defining property of excited states is their quantum number, especially for Π_u , which transforms oddly under $\mathcal{P} \circ \mathcal{C}$ and each $\pi/2$ rotation is assigned with a phase shift of $e^{i\pi/2}$. Thus, the complex phase of each path plays as important a role as the magnitude. Therefore, in addition to the variation of the opacity, we employ a color code indicating the complex phase of the paths. This complicates the picture, but it still provides an impression of what the neural network has learned. For example, if the phases do not matter (as in the Σ_g^+ state), we will not be able to identify regions dominated by a specific phase since all possible phases are summed. In contrast, if the phase matters, we can identify regions of a specific phase since paths with a comparable phase, and hence, a similar color, are summed.

Figure 17 illustrates the cumulative paths of the first two excited states from three different perspectives: The first view is from an off-axis side angle, the second is an on-axis

view orthogonal to the separation axis, and the third perspective is from above, along the separation axis.

Considering the $n = 1$ excited state, from the top view, we can clearly identify four regions where different colors predominate with continuous transitions. The right top region is dominated by green, indicating a complex phase of $\sim\pi/2$, the lower top by red, indicating a complex phase of $\sim 2\pi/\sim 0$, the lower left by purple, indicating a complex phase of $\sim 3\pi/2$, and the top left by turquoise or blue, indicating a complex phase of $\sim\pi$. The side views show that the phase is preserved over the full separations, demonstrating that it is not randomly distributed.

These observations demonstrate that the phase shifts accompanying $\pi/2$ rotations are approximately $e^{i\pi/2}$, as expected for Π_u states. Paths that are reflected at the separation axis exhibit a phase shift of -1 , which corresponds to an odd quantum number for the $(\mathcal{P} \circ \mathcal{C})$ transformation. This is the same observation we had in Fig. 12 for the Π_u state, apart from a global phase difference ambiguity.

When we compare these observations with the $n = 2$ excited state, we see green and red regions interchanged by reflecting along the horizontal axis and, so, reach the same conclusion. The two states are apparently related by the parity transformation, which results in the doublet of the Π_u state. Thus, these findings make more vivid the

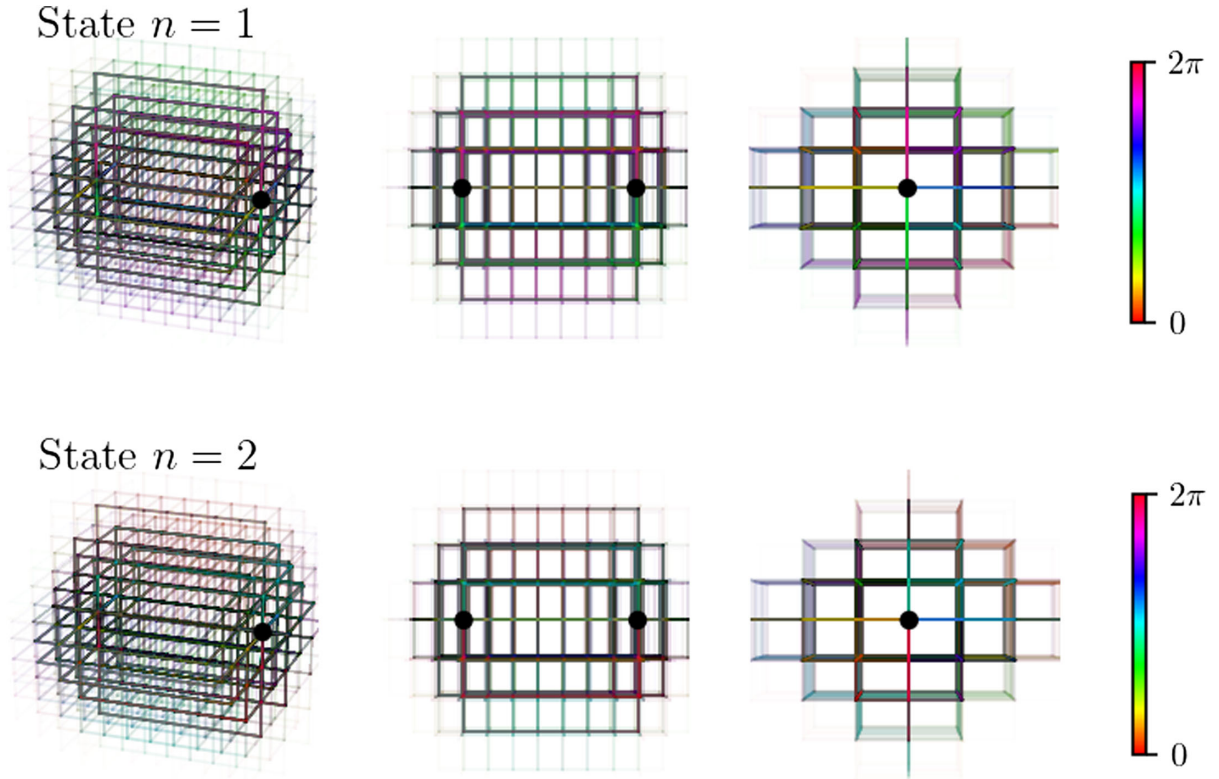


FIG. 17. Visualizations of the neural-network-optimized operator for the first two excited states (top, $n = 1$; bottom, $n = 2$) of the quark-antiquark pair for a particle separation of $r = 7$, shown from three different perspectives. The paths are color coded by the complex phases of their coefficients in the final sum, and the opacity is proportional to the absolute value of these coefficients.

evidence that the neural network has automatically learned the Π_u state.

V. SUMMARY, CONCLUSIONS, AND OUTLOOK

We have developed a new method using neural networks to enhance Wilson loop measurements in a realistic lattice QCD setup. We have introduced gauge-equivariant layers and established a loss function that optimizes for the ground state in a Euclidean correlator. The trained neural network performs as well as, or even better than, Coulomb-gauge Wilson lines, while providing a gauge-invariant setup. This achievement is significant and beneficial for contexts where the Coulomb gauge is not easily applicable.

Further, we have demonstrated that neural-network-based Wilson loops can strikingly improve ground-state expectation values, such as the direct measurement of static forces. We have also explored combining these neural networks with other algorithms, specifically by integrating the multi-level algorithm, to further enhance their application.

Finally, our neural-network method can find the tower of excited states without explicitly specifying the quantum numbers. We found that the static energies of the first excited states exhibit characteristics consistent with the Π_u hybrid state, indicating that the network has successfully identified the Π_u state. This result confirms that the first excited state is indeed the Π_u hybrid state.

Overall, this study provides a foundation for future studies. Some of them might follow directly from this work, and others are extensions of this method to other systems.

A natural next step is to extend this method to off-axis separations to optimize both ground and excited states and to perform high-precision lattice-QCD calculations of, for instance, hybrid static energies and other ground-state expectation values. The combination with multilevel has shown a promising approach to reach high precision. Other combinations with gradient flow or HYP smearing in unquenched theories are also applicable. Furthermore, tailored networks that obey specific quantum numbers can be developed, opening the possibilities of this method for more specialized cases in the future.

Another direction is measuring Wilson loops at nonzero temperature. The attainment of the effective-mass plateau at shorter times may help address the challenge of extracting the static energy from the Wilson loop [52,53].

In addition to these points, several open questions remain. Further research should clarify the extent to which a neural network can improve the Wilson loop (especially for larger r), identify the specific quantum numbers it learns, and explore more effective visualization techniques to illustrate what it has learned.

In the long term, this method can be extended to the generalized Wilson loops required for Born-Oppenheimer effective field theories (BOEFTs) [6,17,19] that describe tetra- and pentaquark systems. We believe our approach

could be particularly convenient for studying these generalized Wilson loops. Notably, the exotic XYZ quarkonia states have recently garnered significant attention in both lattice field theory and experimental studies. The BOEFT framework allows the direct incorporation of lattice input for static energies into phenomenological analyses, providing a simpler, alternative way to study scattering problems in lattice QCD. Wilson-loop-type objects also show up for glueball and gluelump operators. The operators consist of local Wilson loops with specific quantum numbers, so a similar technology can be adapted for this study.

Additionally, Wilson loops also play a role in the low-energy factorization of long-distance matrix elements in quarkonium NRQCD, using strongly coupled pNRQCD [54,55]. This method may enable the first lattice-QCD calculations of quarkonium production, potentially having a significant impact on LHC experiments.

Since the overall structure of the neural network—gauge-equivariant layers, loss functions for training, and optimization on a subset of configurations—is valid for all types of Euclidean correlators, we can also explore neural-network approaches for hadronic states. This can be incorporated into current computational efforts to compute parton distribution functions, hadron vacuum polarizations, and hadron structures.

ACKNOWLEDGMENTS

We thank Tom Magorsch, Simon Pfahler, Marcel Rodekamp, and Marc-Philipp Tillschneider for their useful discussions and inputs. The authors gratefully acknowledge the scientific support and resources of the AI service infrastructure LRZ AI Systems provided by the Leibniz Supercomputing Centre (LRZ) of the Bavarian Academy of Sciences and Humanities (BAdW), funded by Bayerisches Staatsministerium für Wissenschaft und Kunst (StMWK). J. M.-S. acknowledges support by the Munich Data Science Institute (MDSI) at the Technical University of Munich (TUM) via the Linde/MDSI Doctoral Fellowship program. V. B. conducted part of this work in the MIT Center for Theoretical Physics—A Leinweber Institute and acknowledges support from the MIT Department of Physics. N. B. acknowledges the Advanced ERC Grant No. ERC-2023-ADG-Project EFT-XYZ. This document was prepared by the TUMQCD Collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, Office of High Energy Physics HEP User Facility. Fermilab is managed by Fermi Forward Discovery Group, LLC, acting under Contract No. 89243024CSC000002.

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

- [1] K. G. Wilson, Confinement of quarks, *Phys. Rev. D* **10**, 2445 (1974).
- [2] T. Appelquist, M. Dine, and I. J. Muzinich, Static limit of quantum chromodynamics, *Phys. Rev. D* **17**, 2074 (1978).
- [3] N. Brambilla, A. Pineda, J. Soto, and A. Vairo, Infrared behavior of the static potential in perturbative QCD, *Phys. Rev. D* **60**, 091502 (1999).
- [4] N. Brambilla, A. Pineda, J. Soto, and A. Vairo, Potential NRQCD: An effective theory for heavy quarkonium, *Nucl. Phys. B* **566**, 275 (2000).
- [5] N. Brambilla, A. Pineda, J. Soto, and A. Vairo, Effective field theories for heavy quarkonium, *Rev. Mod. Phys.* **77**, 1423 (2005).
- [6] M. Berwein, N. Brambilla, A. Mohapatra, and A. Vairo, Hybrids, tetraquarks, pentaquarks, doubly heavy baryons, and quarkonia in Born-Oppenheimer effective theory, *Phys. Rev. D* **110**, 094040 (2024).
- [7] N. Brambilla, R. L. Delgado, A. S. Kronfeld, V. Leino, P. Petreczky, S. Steinbeißer, A. Vairo, and J. H. Weber (TUMQCD Collaboration), Static energy in $(2 + 1 + 1)$ -flavor lattice QCD: Scale setting and charm effects, *Phys. Rev. D* **107**, 074503 (2023).
- [8] A. Vairo, Strong coupling from the QCD static energy, *Mod. Phys. Lett. A* **31**, 1630039 (2016).
- [9] N. Brambilla, V. Leino, O. Philipsen, C. Reisinger, A. Vairo, and M. Wagner, Lattice gauge theory computation of the static force, *Phys. Rev. D* **105**, 054514 (2022).
- [10] N. Brambilla, V. Leino, J. Mayer-Staudte, and A. Vairo, Static force from generalized Wilson loops on the lattice using the gradient flow, *Phys. Rev. D* **109**, 114517 (2024).
- [11] E. Eichten and F. Feinberg, Spin dependent forces in QCD, *Phys. Rev. D* **23**, 2724 (1981).
- [12] P. de Forcrand and J. D. Stack, Spin dependent potentials in SU(3) lattice gauge theory, *Phys. Rev. Lett.* **55**, 1254 (1985).
- [13] M. Campostrini, K. Moriarty, and C. Rebbi, Monte Carlo calculation of the spin dependent potentials for heavy quark spectroscopy, *Phys. Rev. Lett.* **57**, 44 (1986).
- [14] A. Barchielli, E. Montaldi, and G. M. Prosperi, On a systematic derivation of the quark-antiquark potential, *Nucl. Phys. B* **296**, 625 (1988); **B303**, 752(E) (1988).
- [15] N. Brambilla, A. Pineda, J. Soto, and A. Vairo, QCD potential at $O(1/m)$, *Phys. Rev. D* **63**, 014023 (2001).
- [16] A. Pineda and A. Vairo, QCD potential at $O(1/m^2)$: Complete spin dependent and spin independent result, *Phys. Rev. D* **63**, 054007 (2001); *Phys. Rev. D* **64**, 039902(E) (2001).
- [17] M. Berwein, N. Brambilla, J. Tarrús Castellà, and A. Vairo, Quarkonium hybrids with nonrelativistic effective field theories, *Phys. Rev. D* **92**, 114019 (2015).
- [18] S. Capitani, O. Philipsen, C. Reisinger, C. Riehl, and M. Wagner, Precision computation of hybrid static potentials in SU(3) lattice gauge theory, *Phys. Rev. D* **99**, 034502 (2019).
- [19] J. Soto and J. Tarrús Castellà, Nonrelativistic effective field theory for heavy exotic hadrons, *Phys. Rev. D* **102**, 014012 (2020); *Phys. Rev. D* **110**, 099901(E) (2024).
- [20] C. Schlosser and M. Wagner, Hybrid static potentials in SU(3) lattice gauge theory at small quark-antiquark separations, *Phys. Rev. D* **105**, 054503 (2022).
- [21] M. Eichberg and M. Wagner, Computing $1/m_Q$ and $1/m_Q^2$ corrections to the static potential with lattice gauge theory using gradient flow, *Proc. Sci., LATTICE2024* (2025) 117 [arXiv:2411.11640].
- [22] M. Favoni, A. Ipp, D. I. Müller, and D. Schuh, Lattice gauge equivariant convolutional neural networks, *Phys. Rev. Lett.* **128**, 032003 (2022).
- [23] Y. Nagai, H. Ohno, and A. Tomiya, CASK: A gauge covariant transformer for lattice gauge theory, *Proc. Sci., LATTICE2024* (2025) 030 [arXiv:2501.16955].
- [24] W. Detmold, G. Kanwar, M. L. Wagman, and N. C. Warrington, Path integral contour deformations for noisy observables, *Phys. Rev. D* **102**, 014514 (2020).
- [25] W. Detmold, G. Kanwar, H. Lamm, M. L. Wagman, and N. C. Warrington, Path integral contour deformations for observables in SU(N) gauge theory, *Phys. Rev. D* **103**, 094517 (2021).
- [26] Y. Lin, W. Detmold, G. Kanwar, P. E. Shanahan, and M. L. Wagman, Signal-to-noise improvement through neural network contour deformations for 3D SU(2) lattice gauge theory, *Proc. Sci., LATTICE2023* (2024) 043 [arXiv:2309.00600].
- [27] K. Holland, A. Ipp, D. I. Müller, and U. Wenger, Machine-learned renormalization-group-improved gauge actions and classically perfect gradient flows, *Phys. Rev. Lett.* **136**, 031901 (2026).
- [28] G. Kanwar, Flow-based sampling for lattice field theories, *Proc. Sci., LATTICE2023* (2024) 114 [arXiv:2401.01297].
- [29] A. Tomiya, Machine learning for lattice QCD, *J. Phys. Soc. Jpn.* **94**, 031006 (2025).
- [30] M. Lüscher and P. Weisz, Definition and general properties of the transfer matrix in continuum limit improved lattice gauge theories, *Nucl. Phys. B*, *Proc. Suppl.* **63**, 326 (1984).
- [31] G. S. Bali, K. Schilling, and A. Wachter, Complete $O(v^2)$ corrections to the static interquark potential from SU(3) gauge theory, *Phys. Rev. D* **56**, 2566 (1997).
- [32] K. J. Juge, J. Kuti, and C. J. Morningstar, Gluon excitations of the static quark potential and the hybrid quarkonium spectrum, *Nucl. Phys. B, Proc. Suppl.* **63**, 326 (1998).
- [33] K. J. Juge, J. Kuti, and C. J. Morningstar, *Ab initio* study of hybrid $b\bar{g}b$ mesons, *Phys. Rev. Lett.* **82**, 4400 (1999).
- [34] K. J. Juge, J. Kuti, and C. Morningstar, Fine structure of the QCD string spectrum, *Phys. Rev. Lett.* **90**, 161601 (2003).
- [35] R. Höllwieser, F. Knechtli, T. Korzec, M. J. Peardon, L. Struckmeier, and J. A. Urrea-Niño, Hybrid static potentials and gluelumps on $N_f = 3 + 1$ ensembles, *Proc. Sci., LATTICE2024* (2025) 102 [arXiv:2501.15670].
- [36] Y. Nagai and A. Tomiya, Gauge covariant neural network for quarks and gluons, *Phys. Rev. D* **111**, 074501 (2025).
- [37] C. Lehner and T. Wettig, Gauge-equivariant neural networks as preconditioners in lattice QCD, *Phys. Rev. D* **108**, 034503 (2023).
- [38] M. Albanese *et al.* (APE Collaboration), Glueball masses and string tension in lattice QCD, *Phys. Lett. B* **192**, 163 (1987).
- [39] A. H. Al-Mohy and N. J. Higham, Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation, *SIAM J. Matrix Anal. Appl.* **30**, 1639 (2009).
- [40] P. Bader, S. Blanes, and F. Casas, Computing the matrix exponential with an optimized Taylor polynomial approximation, *Mathematics* **7**, 1174 (2019).

- [41] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, *Science* **355**, 602 (2017).
- [42] C. DeTar *et al.* (MILC Collaboration), MILC code, <https://web.physics.utah.edu/~detar/milc.html> (2013–2016).
- [43] I. Loshchilov and F. Hutter, Decoupled weight decay regularization, in *2019 International Conference on Learning Representations*, edited by T. Sainath *et al.* (2019), [arXiv:1711.05101](https://arxiv.org/abs/1711.05101).
- [44] A. Hasenfratz and F. Knechtli, Flavor symmetry and the static potential with hypercubic blocking, *Phys. Rev. D* **64**, 034504 (2001).
- [45] R. Sommer, A new way to set the energy scale in lattice gauge theories and its applications to the static force and α_s in SU(2) Yang-Mills theory, *Nucl. Phys. B* **411**, 839 (1994).
- [46] C. W. Bernard, T. Burch, K. Orginos, D. Toussaint, T. A. DeGrand, C. E. DeTar, S. A. Gottlieb, U. M. Heller, J. E. Hetrick, and B. Sugar, Static quark potential in three flavor QCD, *Phys. Rev. D* **62**, 034503 (2000).
- [47] A. Vairo, A low-energy determination of α_s at three loops, *EPJ Web Conf.* **126**, 02031 (2016).
- [48] M. Lüscher and P. Weisz, Locality and exponential error reduction in numerical lattice gauge theory, *J. High Energy Phys.* **09** (2001) 010.
- [49] C. Michael and I. Teasdale, Extracting glueball masses from lattice QCD, *Nucl. Phys. B* **215**, 433 (1983).
- [50] A. S. Kronfeld, Improved methods for computing masses from numerical simulations, *Nucl. Phys. B, Proc. Suppl.* **17**, 313 (1990).
- [51] M. Lüscher and U. Wolff, How to calculate the elastic scattering matrix in two-dimensional quantum field theories by numerical simulation, *Nucl. Phys. B* **339**, 222 (1990).
- [52] A. Bazavov, D. Hoying, R. N. Larsen, S. Mukherjee, P. Petreczky, A. Rothkopf, and J. H. Weber (HotQCD Collaboration), Unscreened forces in the quark-gluon plasma?, *Phys. Rev. D* **109**, 074504 (2024).
- [53] S. Ali, D. Bala, O. Kaczmarek, and Pavan (HotQCD Collaboration), Thermal static potential and pseudoscalar quarkonium spectral functions from (2 + 1)-flavor lattice QCD, *Phys. Rev. D* **112**, 054510 (2025).
- [54] N. Brambilla, H. S. Chung, and A. Vairo, Inclusive production of heavy quarkonia in pNRQCD, *J. High Energy Phys.* **09** (2021) 032.
- [55] N. Brambilla, H. S. Chung, A. Vairo, and X.-P. Wang, Inclusive production of J/ψ , $\psi(2S)$, and Υ states in pNRQCD, *J. High Energy Phys.* **03** (2023) 242.