Improving Convergence of Optimization using Teleportation Techniques

George Ma **Yixiao Huang**

Abstract

Optimization problems often involve objective functions with inherent symmetries. For instance, the Rosenbrock function exhibits rotational symmetry, while the parameters of ReLU neural networks possess positive scale-invariance symmetry. These symmetries are typically represented by groups, where different parameter configurations within the same group orbit result in identical loss values. In this work, we accelerate optimization by transporting parameters to equivalent configurations that maintain the same loss but exhibit steeper gradients. We extend this teleportation concept beyond symmetry groups with an approach called level-set teleportation, which guarantees that the teleported parameters achieve both reduced loss values and steeper gradients. Theoretically, we prove that these methods improve the convergence rate of optimization. Experiments on synthetic objective functions and neural network parameter optimization demonstrate significant acceleration in convergence through both symmetry-based and non-symmetry-based teleportation techniques.

1 Introduction

Optimization plays a pivotal role in numerous fields, ranging from machine learning to scientific computing and engineering. Despite the extensive research and advancements in optimization algorithms, certain challenges persist. One common bottleneck is the occurrence of plateaus in the optimization landscape—regions where the gradient magnitude is small, causing the optimization process to stagnate. This phenomenon significantly slows convergence, particularly in high-dimensional problems or in the training of deep neural networks.

To address this issue, we propose the following strategy: teleporting parameters to escape plateaus and accelerate convergence following [9, 4]. The core idea is to leverage properties of the optimization landscape to identify alternative parameter configurations that either maintain equivalent loss values but exhibit steeper gradients or directly result in reduced loss. By relocating parameters to such favorable regions, we enable faster progress during optimization while adhering to the computational constraints of standard training routines.

Our contributions can be summarized as follows:

- *Symmetry Teleportation*. We exploit symmetry properties inherent in optimization problems to transport parameters within equivalent regions of the optimization landscape. For example, rotational symmetry in the Rosenbrock function and positive scale-invariance symmetry in ReLU neural networks allow us to relocate parameters to configurations with steeper gradients while preserving loss values.
- *Level-Set Teleportation*. Extending beyond symmetry-based approaches, we implement level-set teleportation. This method identifies parameter configurations that not only exhibit steeper gradients but also reduce the loss directly, without relying on symmetry properties.
- *Empirical Validation*. We successfully reproduce the algorithms presented in [8, 4] and conduct extensive experiments on both synthetic objective functions and neural network

Preprint. Under review.

training tasks. Our results demonstrate that symmetry teleportation and level-set teleportation consistently enhance convergence rates. Under the same number of training steps and equivalent computational budgets, these techniques achieve superior optimization performance compared to baseline methods.

By combining theoretical insights with empirical results, our work highlights the potential of teleportation techniques to mitigate the challenges posed by plateaus in optimization landscapes. This approach offers a general framework that can be integrated into existing optimization algorithms, paving the way for more efficient and robust solutions in complex applications.

2 Theory

2.1 Symmetry Teleportation

Given loss function $\mathcal{L}$, there may be multiple sets of parameters with the same loss:

$$\mathcal{L}(\boldsymbol{w}) = \mathcal{L}(g \cdot \boldsymbol{w}), \quad \forall g \in G, \forall \boldsymbol{w} \in \mathbb{R}^d,$$

where G is a group acting on the parameter space.

For instance, ReLU neural networks have positive scale invariance. As illustrated in Figure 1, if the incoming weights of a hidden node with ReLU activation are multiplied by a positive constant c and the outgoing weights are divided by c, the neural network with the new weights will generate exactly the same output as the old one for an arbitrary input [3].

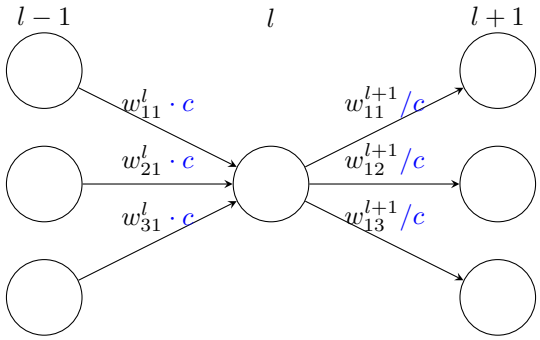


Figure 1: Illustration of the positive scale invariance of ReLU neural networks.

At each optimization step, we can choose a group element g to maximize the magnitude of the gradient [9]:

$$g = \arg\max_{g \in G} \|\nabla \mathcal{L}(g \cdot \boldsymbol{w})\|^2,$$

and conduct gradient descent on $g \cdot \boldsymbol{w}$. In theory, we can prove that this accelerates convergence of SGD, as well as other optimization algorithms. For SGD, the following theorem from [9] shows the improvement in convergence speed using symmetry teleportation:

Proposition 2.1 ([9]). *Let $\mathcal{L}(\boldsymbol{w}, \xi)$ be β-smooth and let*

$$\sigma^2 := \mathcal{L}(\boldsymbol{w}^*) - \mathbb{E}\left[\inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi)\right].$$

Consider the iterates $\boldsymbol{w}^t$ given by $\boldsymbol{w}^{t+1} = g^t \cdot \boldsymbol{w}^t - \eta \nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t, \xi^t)$ where $g^t \in \arg\max_{g \in G} \|\nabla \mathcal{L}(g \cdot \boldsymbol{w}^t)\|^2$, which we assume exists. If $\eta = \frac{1}{\beta\sqrt{T-1}}$ then

$$\min_{t=0,\ldots,T-1} \mathbb{E}\left[\max_{g \in G} \|\nabla \mathcal{L}(g \cdot \boldsymbol{w}^t)\|^2\right] \leq \frac{2\beta}{\sqrt{T-1}} \mathbb{E}\big[\mathcal{L}(\boldsymbol{w}^0) - \mathcal{L}(\boldsymbol{w}^*)\big] + \frac{\beta\sigma^2}{\sqrt{T-1}}, \tag{1}$$

where the expectations is the total expectation with respect to the data ξ^t for $t = 0, \ldots, T-1$. This is an improvement over vanilla SGD, for which we would have instead that

$$\min_{t=0,\ldots,T-1} \mathbb{E}\big[\|\nabla \mathcal{L}(\boldsymbol{w}^t)\|^2\big] \leq \frac{2\beta}{\sqrt{T-1}} \mathbb{E}\big[\mathcal{L}(\boldsymbol{w}^0) - \mathcal{L}(\boldsymbol{w}^*)\big] + \frac{\beta\sigma^2}{\sqrt{T-1}}.$$

In practice, we are interested in the symmetry groups of neural networks with different activation functions. Consider a k-layer neural network, where the hidden dimension of the i-th layer is n_i, and the activation function of the i-th layer is σ_{n_i}. For any $0 \le i < k$, define

$$G_{\sigma_{n_i}} \coloneqq \{\boldsymbol{A} \in \mathrm{GL}_{n_i}(\mathbb{R}) \mid \exists \boldsymbol{B} \in \mathrm{GL}_{n_i}(\mathbb{R}) \text{ s.t. } \sigma_{n_i} \circ \boldsymbol{A} = \boldsymbol{B} \circ \sigma_{n_i}\}.$$

The symmetry groups of multi-layer neural networks with different activation functions are shown in Table 1, where $\phi_\sigma(\boldsymbol{A})$ denotes the matrix $\boldsymbol{B}$ in the definition of $G_{\sigma_{n_i}}$.

Table 1: Symmetry groups of multi-layer neural networks with different activation functions [2]. Here $\boldsymbol{P} \in \Sigma_n$ is a permutation matrix, $\boldsymbol{D}$ is a diagonal matrix, abs denotes entry-wise absolute value, and $\boldsymbol{A}^{\odot d}$ denotes the entry-wise d-th power.

Activation	G_{σ_n}	$\phi_\sigma(\boldsymbol{A})$
$\sigma(x) = x$ (identity)	$\mathrm{GL}_n(\mathbb{R})$	$\boldsymbol{A}$
$\sigma(x) = \frac{e^x}{1+e^x}$	Σ_n (permutation matrices)	$\boldsymbol{A}$
$\sigma(x) = \mathrm{ReLU}(x)$	Matrices $\boldsymbol{PD}$, where $\boldsymbol{D}$ has positive entries	$\boldsymbol{A}$
$\sigma(x) = \mathrm{LeakyReLU}(x)$	Same as ReLU as long as negative slope $\neq 1$	$\boldsymbol{A}$
$\sigma(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ (RBF)	Matrices $\boldsymbol{PD}$, where $\boldsymbol{D}$ has entries in $\{\pm 1\}$	$\mathrm{abs}(\boldsymbol{A})$
$\sigma(x) = x^d$ (polynomial)	Matrices $\boldsymbol{PD}$, where $\boldsymbol{D}$ has non-zero entries	$\boldsymbol{A}^{\odot d}$

In Section 4.1.2, we conduct symmetry teleportation during the optimization of a ReLU neural network, and show that we can achieve smaller loss both within the same number of steps and the same amount of training time.

2.2 Level-Set Teleportation

The limitation of symmetry teleportation is clear as it does not apply to non-symmetric problems. One way to relieve this limitation is to consider a more general objective called level-set teleportation as in [4]:

$$w_k^* = \arg\max_w \frac{1}{2}\|\nabla \mathcal{L}(w)\|^2, \quad \text{s.t. } \mathcal{L}(w) \le \mathcal{L}(w_k). \tag{2}$$

One can apply standard SQP tools [5] to solve this sub-level set optimization problem by approximating the problem using Taylor's theorem as in Equation 3:

$$\begin{aligned} w_k^* = \arg\max_w \left\langle \nabla^2 \mathcal{L}(w_k)\nabla \mathcal{L}(w_k), w - w_k \right\rangle + \frac{1}{2}(w-w_k)^{\mathrm{T}} B_k (w - w_k), \\ \text{s.t. } \nabla \mathcal{L}(w_k)^{\mathrm{T}}(w - w_k) \le 0, \end{aligned} \tag{3}$$

where B_k is the Hessian of $\frac{1}{2}\|\nabla \mathcal{L}(w)\|^2$. As a result, the objective includes a third-order derivative of $\mathcal{L}(w)$, making the computation intractable for large-scale problems. To proceed, we follow [4] to consider an iterative method [1] where the initial value is $x_0 = w_k$ and denote the iterates as x_t:

$$x_{t+1} = \arg\max_{x \in \tilde{S}_k(x_t)} \left\{ \frac{1}{2}\log \|\nabla \mathcal{L}(x_t)\|_2^2 + \left\langle \frac{\nabla^2 \mathcal{L}(x_t)\nabla \mathcal{L}(x_t)}{\|\nabla \mathcal{L}(x_t)\|_2^2}, x - x_t \right\rangle - \frac{1}{\rho_t}\|x - x_t\|_2^2 \right\}, \tag{4}$$

where $\tilde{S}_k(x_t) = \{x \colon \mathcal{L}(x_t) + \nabla \mathcal{L}(x_t)^{\mathrm{T}}(x - x_t) \le \mathcal{L}(w_k)\}$. This can be viewed as SQP formulation on the logarithm of the objective function. As we will see in the following analysis, the objective can be interpreted as projected gradient descent methods [6] with a linearized constraint.

Proposition 2.2 (Proposition 3.1 in [4]). *Let $q_t = \nabla^2 \mathcal{L}(x_t)\nabla \mathcal{L}(x_t)$, $g_t = \|\nabla \mathcal{L}(x_t)\|_2^2$, the solution to Equation 4 is given by:*

$$\begin{aligned} v_t &= -\left(\rho_t \frac{\langle \mathcal{L}(x_t), q_t\rangle}{g_t} + \mathcal{L}(x_t) - \mathcal{L}(w_k)\right)_+ \nabla \mathcal{L}(x_t), \\ x_{t+1} &= x_t + (\rho_t q_t + v_t)/g_t, \end{aligned}$$

where $(u)_+ = \max\{0, u\}$ and $\rho_t > 0$ is a hyperparameter.

[1] The rest of this section basically follows [4] while we corrected several mistakes (highlighted in purple) in their original analysis.

The complete proof is given in Appendix B.1.

As indicated in the previous proposition, ρ_t is a hyperparameter that needs to be tuned. To solve this problem, we conduct a line-search based on the Armijo-type condition following standard practice for SQP [5]:

$$\phi_\gamma(x_{t+1}) \le \phi_\gamma(x_t) + \frac{1}{2} D_\phi(x_t, x_{t+1} - x_t),$$

where $\phi_\gamma(x_t, w_k) = -\frac{1}{2}\|\nabla \mathcal{L}(x_t)\|_2^2 + \gamma(\mathcal{L}(x_t) - \mathcal{L}(w_k))_+$ is a merit function, $\gamma_t > 0$ is the penalty strength, and $D_\phi(x_t, x_{t+1} - x_t)$ is the directional derivative of ϕ_γ in direction $x_{t+1} - x_t$ evaluated at x_t. Below we prove that setting γ sufficiently large gives a descent direction of ϕ_γ.

Lemma 2.3 (Lemma B.1 in [4]). *The directional derivative of the merit function satisfies*

$$D_\phi(x_t, x_{t+1} - x_t) \le -\frac{\langle q_t, v_t\rangle + \rho_t \|q_t\|_2^2}{g_t} - \gamma\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+.$$

As a result, if $\mathcal{L}(x_t) - \mathcal{L}(w_k) > 0$ and

$$\gamma \ge \max\left\{0, -\frac{\langle q_t, v_t\rangle}{g_t\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)}\right\},$$

then $x_{t+1} - x_t$ is a descent direction for ϕ_γ at x_t. When $\mathcal{L}(x_t) - \mathcal{L}(w_k) \le 0$, if $\rho_t \ge -\langle q_t, v_t\rangle / \|q_t\|_2^2$, $x_{t+1} - x_t$ is a descent direction.

The proof basically follows [4] except that we also discuss the case when $\mathcal{L}(x_t) - \mathcal{L}(w_k) \le 0$.

Proposition 2.4 (Proposition 3.2 in [4]). *If $\mathcal{L}(x_t) - \mathcal{L}(w_k) > 0$ and*

$$\gamma \ge \max\left\{0, -\frac{\langle q_t, v_t\rangle}{g_t\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)}\right\},$$

then $x_{t+1} - x_t$ is a descent direction of ϕ_{γ_t} and the line-search condition simplifies to

$$\phi_{\gamma_t}(x_{t+1}) \le -\frac{1}{2} g_t - \frac{\langle q_t, v_t\rangle + \rho_t \|q_t\|_2^2}{g_t}.$$

Remark 1. [4] didn't provide analysis on the scenario when $\mathcal{L}(x_t) - \mathcal{L}(w_k) \le 0$, e.g. this happens in the first iteration of teleportation where $x_t = w_k$. In practice, we set γ_t to a sufficiently large number, such as 10^9, to keep x_{t+1} inside the feasible region and accept the ρ_t as long as it gives a descent direction, i.e., $\phi_{\gamma_t}(x_{t+1}) < \phi_{\gamma_t}(x_t)$.

2.3 Connection with Newton's method

Proposition 2.1 provides theoretical evidence for improved convergence of symmetry teleportation. In this part, we show that the solution to the general teleportation objective, i.e., Equation 2, has a nice connection with Newton's method. By the KKT conditions, we have

$$\nabla^2 \mathcal{L}(w_k^+) \nabla \mathcal{L}(w_k) = \lambda_k \nabla \mathcal{L}(w_k^+),$$
$$\mathcal{L}(w_k^+) \le \mathcal{L}(w_k) \text{ and } \lambda_k(\mathcal{L}(w_k^+) - \mathcal{L}(w_k)) = 0,$$

where $\lambda_k \ge 0$ is the Lagrangian multiplier. Solving the equations above, we get

$$\nabla \mathcal{L}(w_k^+) = \lambda_k \big(\nabla^2 \mathcal{L}(w_k^+)\big)^\dagger \nabla \mathcal{L}(w_k^+),$$

where $(\nabla^2 \mathcal{L}(w_k^+))^\dagger$ is the pseudo-inverse of the Hessian matrix. As a result, if $\lambda_k \ne 0$, the first gradient descent after teleportation is equivalent to a one-step Newton's method, which explains why teleportation can improve the convergence of gradient descent.

3 Algorithms

In this section, we give an overview of the gradient descent method with symmetry/level-set teleportation, which can be summarized in Algorithm 1.

Algorithm 1 Gradient descent with teleportation

Require: Parameters $\boldsymbol{w}$, learning rate η, teleportation epochs $\mathcal{T} \subseteq \{1, \dots, K\}$
Ensure: The optimized parameters $\boldsymbol{w}$
  for $k = 1, \dots, K$ **do**
    if $k \in \mathcal{T}$ **then**
      Conduct teleportation on $\boldsymbol{w}$ with Algorithm 2 or Algorithm 3
    end if
    if line-search **then**
      $g_k \leftarrow \|\nabla \mathcal{L}(\boldsymbol{w})\|^2$
      $\eta \leftarrow \max\{\eta \colon \mathcal{L}(\boldsymbol{w} - \eta \nabla \mathcal{L}(\boldsymbol{w})) \leq \mathcal{L}(\boldsymbol{w}) - \frac{\eta}{2} g_k\}$
    end if
    $\boldsymbol{w} \leftarrow \boldsymbol{w} - \eta \nabla \mathcal{L}(\boldsymbol{w})$
  end for
  return $\boldsymbol{w}$

3.1 Algorithm for Symmetry Teleportation

To implement symmetry teleportation in practice, we can conduct gradient ascent on the norms of gradients of the parameters to find the group element that results in steepest gradient descent direction, and teleport the parameters using that group element. Specifically, let $\boldsymbol{w} = (\boldsymbol{w}_1, \dots, \boldsymbol{w}_d)$ be the parameters that we are trying to optimize, and let $\boldsymbol{G}_0$ be the matrix representation of the initial symmetry group element g_0 that acts on the parameters. To find the group element that results in the steepest gradient descent direction, we first apply $\boldsymbol{G}_0$ to the parameters, and compute the sum of the norms of the gradients of all the parameters: $\boldsymbol{H} = \sum_{i=1}^{d} \|\partial \mathcal{L}(\boldsymbol{G}_0 \cdot \boldsymbol{w}) / \partial (\boldsymbol{G}_0 \cdot \boldsymbol{w}_i)\|^2$, where $\mathcal{L}$ is the loss function. Then, we compute the gradient of $\boldsymbol{H}$ with respect to the initial group action $\boldsymbol{G}_0$, and conduct gradient ascent on $\boldsymbol{G}_0$ to get the group action $\boldsymbol{G}$. We can repeat this process for a number of steps. Finally, we teleport the original parameters to the new parameters $\boldsymbol{G} \cdot \boldsymbol{w} = [\boldsymbol{G} \cdot \boldsymbol{w}_1, \dots, \boldsymbol{G} \cdot \boldsymbol{w}_d]$ which have steeper gradients, and proceed with our optimization algorithm on these new parameters. The algorithm is described in Algorithm 2.

Algorithm 2 Symmetry Teleportation

Require: Parameters $\boldsymbol{w}$, the loss function $\mathcal{L}$, learning rate η_{tel}, teleportation steps n_{tel}
Ensure: The teleported parameters $\boldsymbol{G} \cdot \boldsymbol{w}$ that have steeper gradients
  for $t = 1, \dots, n_{\text{tel}}$ **do**
    Initialize the symmetry group action $\boldsymbol{G}_0$
    Compute the sum of gradient norms $\boldsymbol{H} \leftarrow \sum_{i=1}^{d} \|\partial \mathcal{L}(\boldsymbol{G}_0 \cdot \boldsymbol{w}) / \partial (\boldsymbol{G}_0 \cdot \boldsymbol{w}_i)\|^2$
    Compute the gradient of $\boldsymbol{H}$ with respect to $\boldsymbol{G}_0$: $\Delta \boldsymbol{G} \leftarrow \partial \boldsymbol{H} / \partial \boldsymbol{G}_0$
    Conduct gradient ascent on $\boldsymbol{G}_0$: $\boldsymbol{G} \leftarrow \boldsymbol{G}_0 + \eta_{\text{tel}} \Delta \boldsymbol{G}$
    Teleport the parameters: $\boldsymbol{w} \leftarrow \boldsymbol{G} \cdot \boldsymbol{w} = [\boldsymbol{G} \cdot \boldsymbol{w}_1, \dots, \boldsymbol{G} \cdot \boldsymbol{w}_d]$
  end for
  return the teleported parameters $\boldsymbol{w}$

For gradient descent with K total epochs, we conduct symmetry teleportation on a subset $\mathcal{T}$ of these epochs. The algorithm for gradient descent with symmetry teleportation is shown in Algorithm 1.

In the experiments, for efficiency reasons, we set $|\mathcal{T}| = 1$, which means we only conduct symmetry teleportation on a single epoch. This allows us to accelerate optimization with negligible computational overhead, reaching superior performance to vanilla optimization algorithms within the same amount of time.

3.2 Algorithm for Level-Set Teleportation

The complete algorithm for level-set teleportation is listed in Algorithm 3 in Appendix B.4. Compared to the one in [4], we add a checker on whether $\mathcal{L}(x_t) \leq \mathcal{L}(w_k)$ and apply different conditions for line-search. We use the KKT condition to check whether we can end the teleportation early which is given by $\|\boldsymbol{P}_t q_t\|_2 \leq \epsilon$ and $\mathcal{L}(x_t) - \mathcal{L}(w_k) \leq \delta$ where $\boldsymbol{P}_t$ is the projection to the orthogonal complement of $\nabla \mathcal{L}(x_t)$.

4 Experiments

4.1 Symmetry Teleportation

4.1.1 Convex Optimization

We conduct convex optimization experiments using the following objective function, which is also called Booth function: $f(x, y) = x^2 + 9y^2$, where x, y are the parameters. We observe that the vector $(x, 3y)$ has rotational symmetry, namely, for any rotation matrix $\boldsymbol{Q} \in \mathbb{R}^{2\times 2}$, the parameters $[x, 3y]^{\mathrm{T}}$ and $\boldsymbol{Q}[x, 3y]^{\mathrm{T}}$ correspond to the same value of the objective function. Therefore, we can use the rotational group as the symmetry group. Moreover, this objective function has multiple optimization paths which, under the same learning rate, have different convergence speeds.

Figure 2 shows the level-set plot of the objective function. The optimum point is $(x, y) = (0, 0)$. As shown in Figure 2, both the blue and orange optimization path lead to the optimum point. However, under the same learning rate, the orange path will have higher convergence rate than the blue path. If we start our optimization process near the blue path, we can use symmetry teleportation to teleport to the orange path to accelerate the optimization.

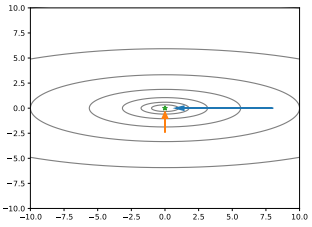
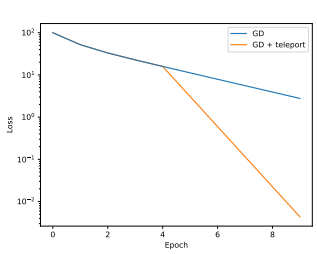
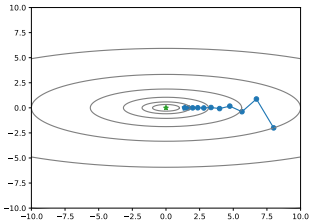


Figure 2: The level-set plot of $f(x, y)$.

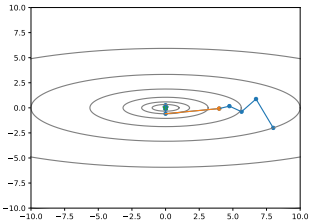
We set the initial point to $(x_0, y_0) = (8, -2)$, which is closer to the slower blue optimization path. We use gradient descent to optimize the objective function for 10 epochs, and only conduct symmetry teleportation at the 4th epoch. The experiment results are shown in Figure 3.



(a) Loss curve.



(b) Level-set plot without teleportation.



(c) Level-set plot with teleportation.

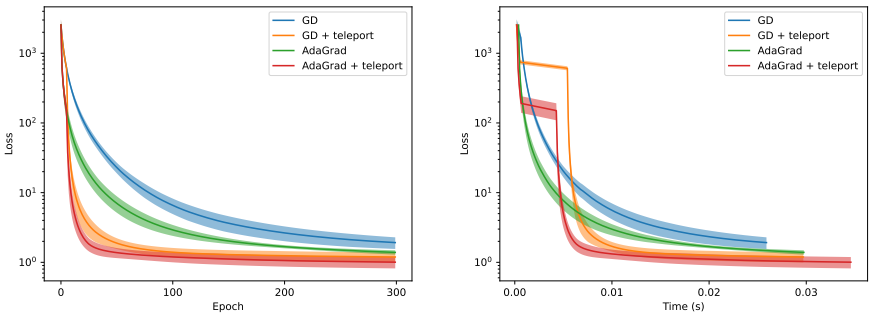
Figure 3: The loss curves and level-set plots of the convex optimization process without and with symmetry teleportation.

The plots of the loss curves and level-sets of the optimization process without and with symmetry teleportation are shown in Figure 3. As shown in Figure 3a, after applying symmetry teleportation at the 4th epoch, the convergence speed of optimization is greatly accelerated, leading to a smaller loss value. The vanilla gradient descent method fails to reach the optimum in 10 epochs, as shown in Figure 3b. However, symmetry teleportation allows the optimization path to teleport from the slower one to the faster one, eventually enabling us to reach the optimum point, as shown in Figure 3c.

4.1.2 Optimization of ReLU Neural Networks

In Section 2.1, we introduced the symmetry groups of neural networks with different activation functions. In particular, ReLU neural networks have positive-scale invariance. In this section, we apply symmetry teleportation to a 3-layer ReLU neural network to demonstrate the performance improvements and efficiency of our approach.

The task is to fit a pair of randomly generated input and output matrices $\boldsymbol{X} \in \mathbb{R}^{5\times 4}, \boldsymbol{Y} \in \mathbb{R}^{8\times 4}$. The ReLU neural network consists of three linear layers without bias, and we denote the parameters of the network as $\boldsymbol{w} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3)$, where $\boldsymbol{w}_1 \in \mathbb{R}^{6\times 5}, \boldsymbol{w}_2 \in \mathbb{R}^{7\times 6}, \boldsymbol{w}_3 \in \mathbb{R}^{8\times 7}$. We train the neural network with gradient descent and AdaGrad [1] for 300 epochs and only conduct symmetry teleportation at the 5th epoch. Each experiment is conducted on 5 different random seeds and the standard errors are reported. The results are shown in Figure 4.



(a) Loss curve with respect to epochs. (b) Loss curve with respect to time.

Figure 4: The loss curve of the ReLU neural network during optimization, with respect to the number of epochs and the training time.

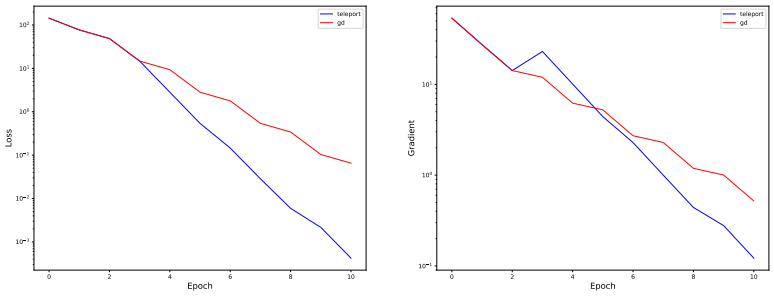
The plots of the loss curve of the ReLU neural network are shown in Figure 4. As shown in Figure 4a, applying symmetry teleportation at the 5th epoch accelerates optimization afterward, allowing us to reach smaller loss values. Since we only conduct symmetry teleportation once, the additional computational overhead is negligible, and as shown in Figure 4b, we also reach better performance within the same amount of training time.

4.2 Level-Set Teleportation

To compare with symmetry teleportation, we also conduct experiments on synthetic test functions and optimizing neural networks.

4.2.1 Optimization of Test Functions

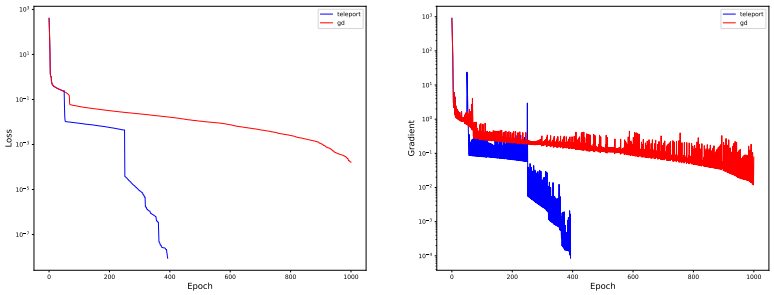
In addition to the Booth function, we also consider optimizing the Rosenbrock function, which is non-convex: $f(x, y) = 100(x^2 - y)^2 + (x - y)^2$. This function has the global minimum at $f(x^*, y^*) = 0$ with $x^* = 1, y^* = 1$. The results of the two test functions are shown in Figure 5 and 6, which show that level-set teleportation can provide comparable performance with symmetry teleportation while enjoying no limitation on the problem structure. Moreover, as shown in Figure 7, calling the teleportation occasionally (once in the Booth function and twice in the Rosenbrock function) is sufficient to give satisfying results. In both of the experiments, we set the maximum teleportation steps to 100 and the maximum line-search steps in teleportation to 50.



(a) Loss curve. (b) Gradient norm curve.

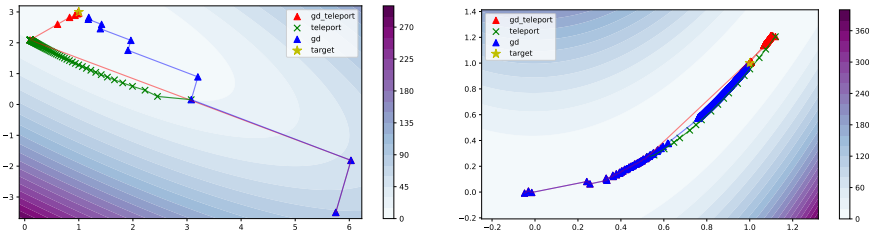
Figure 5: The loss and gradient norm curves of optimizing the Booth function with and without level-set teleportation.

Note that line-search is used to find the optimal step size in the level-set teleportation. As a result, we also apply line-search on gradient descent to ensure a fair comparison. This contrast to the experiments in the symmetry teleportation, which does not use the line-search. We also provide additional results without line-search on gradient descent in Appendix B.5.



(a) Loss curve. (b) Gradient norm curve.

Figure 6: The loss and gradient norm curves of optimizing the Rosenbrock function with and without level-set teleportation.

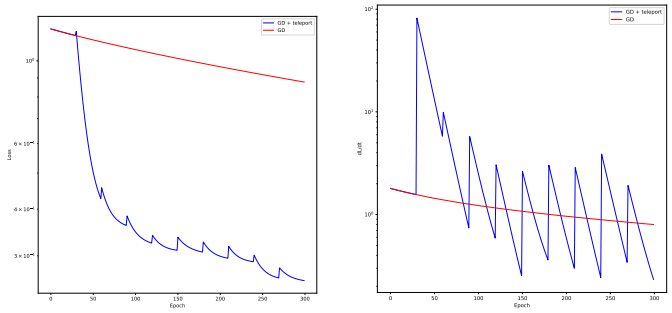


(a) Parameter trajectory of Booth. (b) Parameter trajectory of Rosenbrock.

Figure 7: Parameter trajectory of optimizing the Booth (left) and Rosenbrock (right) functions.

4.2.2 Optimization of Neural Networks

We use a two-layer Softplus network with 50 hidden neurons. The result in Figure 8 shows that level-set teleportation can greatly accelerate the training. Due to the additional cost of teleportation, we defer the experiments on larger networks as a future direction.



(a) Loss curve. (b) Gradient norm curve.

Figure 8: The loss and gradient norm curves of optimizing a two-layer MLPs.

5 Conclusion

We implemented symmetry teleportation and level-set teleportation as effective strategies to address plateaus in optimization landscapes. These methods transport parameters to regions with steeper gradients and lower loss values, accelerating convergence without additional computational overhead. Our theoretical and empirical results demonstrated the significant performance improvements these techniques provide for both synthetic objective functions and neural network training tasks. This work highlights the potential of teleportation methods to enhance optimization efficiency and offers promising directions for future research in large-scale and complex optimization problems.

References

[1] John Duchi, Elad Hazan, and Yoram Singer. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. In *JMLR*, 2011.

[2] Charles Godfrey, Davis Brown, Tegan Emerson, and Henry Kvinge. On the Symmetries of Deep Learning Models and their Internal Representations. In *NeurIPS*, 2022.

[3] Qi Meng, Shuxin Zheng, Huishuai Zhang, Wei Chen, Qiwei Ye, Zhi-Ming Ma, Nenghai Yu, and Tie-Yan Liu. $\mathcal{G}$-SGD: Optimizing ReLU Neural Networks in its Positively Scale-Invariant Space. In *ICLR*, 2019.

[4] Aaron Mishkin, Alberto Bietti, and Robert M Gower. Level Set Teleportation: An Optimization Perspective. *arXiv preprint arXiv:2403.03362*, 2024.

[5] Jorge Nocedal and Stephen J Wright. *Numerical Optimization*. Springer, 1999.

[6] Benjamin Recht and Stephen J. Wright. *Optimization for Modern Data Analysis*. Cambridge University Press, 2019. Preprint available at http://eecs.berkeley.edu/~brecht/opt4mlbook.

[7] Sebastian U Stich. Unified Optimal Analysis of the (Stochastic) Gradient Method. In *CoRR*, 2019.

[8] Bo Zhao, Nima Dehmamy, Robin Walters, and Rose Yu. Symmetry Teleportation for Accelerated Optimization. In *NeurIPS*, 2022.

[9] Bo Zhao, Robert M Gower, Robin Walters, and Rose Yu. Improving Convergence and Generalization Using Parameter Symmetries. In *ICLR*, 2024.

A Proofs for Symmetry Teleportation

A.1 Proof of Proposition 2.1

First we introduce the descent lemma [9].

Lemma A.1 (Descent Lemma)**.** *Let $\mathcal{L}(\boldsymbol{w}, \xi)$ be a β-smooth function. It follows that*

$$\mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{w}, \xi)\|^2\right] \leq 2\beta\big(\mathcal{L}(\boldsymbol{w}) - \mathcal{L}(\boldsymbol{w}^*)\big) + 2\beta\left(\mathcal{L}(\boldsymbol{w}^*) - \mathbb{E}\left[\inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi)\right]\right).$$

Proof. Since $\mathcal{L}(\boldsymbol{w}, \xi)$ is smooth we have that

$$\mathcal{L}(\boldsymbol{z}, \xi) - \mathcal{L}(\boldsymbol{w}, \xi) \leq \left\langle \nabla \mathcal{L}(\boldsymbol{w}, \xi), \boldsymbol{z} - \boldsymbol{w} \right\rangle + \frac{\beta}{2}\|\boldsymbol{z} - \boldsymbol{w}\|^2, \quad \forall \boldsymbol{z}, \boldsymbol{w} \in \mathbb{R}^d.$$

Plugging in

$$\boldsymbol{z} = \boldsymbol{w} - \frac{1}{\beta}\nabla \mathcal{L}(\boldsymbol{w}, \xi),$$

we get

$$\mathcal{L}\big(\boldsymbol{w} - (1/\beta)\nabla \mathcal{L}(\boldsymbol{w}, \xi), \xi\big) \leq \mathcal{L}(\boldsymbol{w}, \xi) - \frac{1}{2\beta}\|\nabla \mathcal{L}(\boldsymbol{w}, \xi)\|^2.$$

Rearranging, we have that

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}^*, \xi) - \mathcal{L}(\boldsymbol{w}, \xi) &= \mathcal{L}(\boldsymbol{w}^*, \xi) - \inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi) + \inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi) - \mathcal{L}(\boldsymbol{w}, \xi) \\
&\leq \mathcal{L}(\boldsymbol{w}^*, \xi) - \inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi) + \mathcal{L}\big(\boldsymbol{w} - (1/\beta)\nabla \mathcal{L}(\boldsymbol{w}, \xi) - \mathcal{L}(\boldsymbol{w}, \xi) \\
&\leq \mathcal{L}(\boldsymbol{w}^*, \xi) - \inf_{\boldsymbol{w}} (\boldsymbol{w}, \xi) - \frac{1}{2\beta}\|\nabla \mathcal{L}(\boldsymbol{w}, \xi)\|^2,
\end{aligned}$$

where the first inequality follows because $\inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi) \leq \mathcal{L}(\boldsymbol{w}, \xi), \forall \boldsymbol{w}$. Rearranging the above and taking expectation gives

$$\begin{aligned}
\mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{w}, \xi)\|^2\right] &\leq 2\mathbb{E}\left[\beta\big(\mathcal{L}(\boldsymbol{w}^*, \xi) - \inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi) + \mathcal{L}(\boldsymbol{w}, \xi) - \mathcal{L}(\boldsymbol{w}^*, \xi)\big)\right] \\
&\leq 2\beta\mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^*, \xi) - \inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi) + \mathcal{L}(\boldsymbol{w}, \xi) - \mathcal{L}(\boldsymbol{w}^*, \xi)\right] \\
&\leq 2\beta\big(\mathcal{L}(\boldsymbol{w}) - \mathcal{L}(\boldsymbol{w}^*)\big) + 2\beta\left(\mathcal{L}(\boldsymbol{w}^*) - \mathbb{E}\left[\inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi)\right]\right).
\end{aligned}$$

□

Then we give the proof of Proposition 2.1 [9].

Proof. First note that if $\mathcal{L}(\boldsymbol{w}, \xi)$ is β-smooth, then $\mathcal{L}(\boldsymbol{w})$ is also a β-smooth function, that is

$$\mathcal{L}(\boldsymbol{z}) - \mathcal{L}(\boldsymbol{w}) - \langle \nabla \mathcal{L}(\boldsymbol{w}), \boldsymbol{z} - \boldsymbol{w} \rangle \leq \frac{\beta}{2}\|\boldsymbol{z} - \boldsymbol{w}\|^2.$$

Using the update rule of $\boldsymbol{w}^t$ with $\boldsymbol{z} = \boldsymbol{w}^{t+1}$ and $\boldsymbol{w} = g^t \cdot \boldsymbol{w}^t$, together with the above equation and the fact that the group action preserves loss, we have that

$$\begin{aligned}
\mathcal{L}(\boldsymbol{w}^{t+1}) &\leq \mathcal{L}(g^t \cdot \boldsymbol{w}^t) + \langle \nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t), \boldsymbol{w}^{t+1} - g^t \cdot \boldsymbol{w}^t \rangle + \frac{\beta}{2}\|\boldsymbol{w}^{t+1} - g^t \cdot \boldsymbol{w}^t\|^2 \\
&= \mathcal{L}(\boldsymbol{w}^t) - \eta_t \langle \nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t), \nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t, \xi^t) \rangle + \frac{\beta \eta_t^2}{2}\|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t, \xi^t)\|^2.
\end{aligned}$$

Taking expectation conditioned on $\boldsymbol{w}^t$, we have that

$$\mathbb{E}_t\left[\mathcal{L})(\boldsymbol{w}^{t+1})\right] \leq \mathcal{L}(\boldsymbol{w}^t) - \eta_t\|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t)\|^2 + \frac{\beta \eta_t^2}{2}\mathbb{E}_t\left[\|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t, \xi^t)\|^2\right].$$

Now since $\mathcal{L}(\boldsymbol{w}, \xi)$ is β-smooth, from Lemma A.1 above we have that

$$\mathbb{E}\left[\|\nabla \mathcal{L}(\boldsymbol{w}, \xi)\|^2\right] \leq 2\beta\big(\mathcal{L}(\boldsymbol{w}) - \mathcal{L}(\boldsymbol{w}^*)\big) + 2\beta\left(\mathcal{L}(\boldsymbol{w}^*) - \mathbb{E}\left[\inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi)\right]\right).$$

Using the above equation with $\boldsymbol{w} = g^t \cdot \boldsymbol{w}^t$ we have that

$$\mathbb{E}_t\left[\mathcal{L}(\boldsymbol{w}^{t+1})\right] \leq \mathcal{L}(\boldsymbol{w}^t) - \eta_t \|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t)\|^2 + \beta^2 \eta_t^2 \left(\mathcal{L}(g^t \cdot \boldsymbol{w}^t) - \mathcal{L}(\boldsymbol{w}^*) + \mathcal{L}(\boldsymbol{w}^*) - \mathbb{E}\left[\inf_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, \xi)\right]\right).$$

Using that $\mathcal{L})(g^t \cdot \boldsymbol{w}^t) = \mathcal{L}(\boldsymbol{w}^t)$, taking full expectation and rearranging terms gives

$$\eta_t \mathbb{E}\left[\|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t)\|^2\right] \leq (1 + \beta^2 \eta_t^2)\mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^t) - \mathcal{L}^*\right] - \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^{t+1}) - \mathcal{L}^*\right] + \beta^2 \eta_t^2 \sigma^2.$$

Now we use a re-weighting trick introduced in [7]. Let $\alpha_t > 0$ be a sequence such that $\alpha_t(1 + \beta^2\eta_t^2) = \alpha_{t-1}$. Consequently if $\alpha_{-1} = 1$ then $\alpha_t = (1 + \beta^2\eta_t^2)^{-(t+1)}$. Multiplying by both sides of the above equation by α_t thus gives

$$\alpha_t \eta_t \mathbb{E}\left[\|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t)\|^2\right] \leq \alpha_{t-1}\mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^t) - \mathcal{L}^*\right] - \alpha_t \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^{t+1}) - \mathcal{L}^*\right] + \alpha_t \beta^2 \eta_t^2 \sigma^2.$$

Summing up from $t = 0, \ldots, T-1$, and using telescopic cancellation, gives

$$\sum_{t=0}^{T-1} \alpha_t \eta_t \mathbb{E}\left[\|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t)\|^2\right] \leq \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^0) - \mathcal{L}^*\right] + \beta^2 \sigma^2 \sum_{t=0}^{T-1} \alpha_t \eta_t^2.$$

Let $A = \sum_{t=0}^{T-1} \alpha_t \eta_t$. Dividing both sides by A gives

$$\begin{aligned} \min_{t=0,\ldots,T-1} \mathbb{E}\left[\|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t)\|^2\right] &\leq \frac{1}{\sum_{t=0}^{T-1} \alpha_t \eta_t} \sum_{t=0}^{T-1} \alpha_t \eta_t \|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t)\|^2 \\ &\leq \frac{\mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^0) - \mathcal{L}^*\right] + \beta^2 \sigma^2 \sum_{t=0}^{T-1} \alpha_t \eta_t^2}{\sum_{t=0}^{T-1} \alpha_t \eta_t}. \end{aligned} \tag{5}$$

Finally, if $\eta_t \equiv \eta$ then

$$\begin{aligned} \sum_{t=0}^{T-1} \alpha_t \eta_t &= \eta \sum_{t=0}^{T-1} (1 + \beta^2 \eta_t^2)^{-(t+1)} = \frac{\eta}{1 + \beta^2 \eta^2} \frac{1 - (1 + \beta^2 \eta^2)^{-T}}{1 - (1 + \beta^2 \eta^2)^{-1}} \\ &= \frac{1 - (1 + \beta^2 \eta^2)^{-T}}{\beta^2 \eta}. \end{aligned}$$

To bound the term with the $-T$ power, we use that

$$(1 + \beta^2 \eta^2)^{-T} \leq \frac{1}{2} \quad \Longrightarrow \quad \frac{\log 2}{\log(1 + \beta^2 \eta^2)} \leq T.$$

To simplify the above expression we can use

$$\frac{x}{1 + x} \leq \log(1 + x) \leq x, \quad \text{for } x \geq -1,$$

thus

$$\frac{\log 2}{\log(1 + \beta^2 \eta^2)} \leq \frac{1 + \beta^2 \eta^2}{\beta^2 \eta^2} \leq T.$$

Using the above we have that

$$\sum_{t=0}^{T-1} \alpha_t \eta_t \geq \frac{1}{2\beta^2 \eta}, \quad \text{for } T \geq \frac{1 + \beta^2 \eta^2}{\beta^2 \eta^2}.$$

Using this lower bound in Equation 5 gives

$$\min_{t=0,\ldots,T-1} \mathbb{E}\left[\|\nabla \mathcal{L}(g^t \cdot \boldsymbol{w}^t)\|^2\right] \leq 2\beta^2 \eta \mathbb{E}\left[\mathcal{L}(\boldsymbol{w}^0) - \mathcal{L}^*\right] + \eta \beta^2 \sigma^2, \quad \text{for } T \geq \frac{1 + \beta^2 \eta^2}{\beta^2 \eta^2}.$$

Now note that

$$T \geq \frac{1 + \beta^2 \eta^2}{\beta^2 \eta^2} \Leftrightarrow \beta^2 \eta^2 (T - 1) \geq 1 \Leftrightarrow \eta \geq \frac{1}{\beta \sqrt{T - 1}}.$$

Thus finally setting $\eta = \frac{1}{\beta\sqrt{T-1}}$ gives the result Equation 1. □

B Proofs for Level-Set Teleportation

B.1 Proof of Proposition 2.2

Proof. Let ρ_t be a hyperparameter to be determined and consider the following unconstrained problem:

$$\bar{x} = \arg\max_{x \in \mathbb{R}^d} \left\{ \left\langle \frac{q_t}{g_t}, x - x_t \right\rangle - \frac{1}{\rho_t} \|x - x_t\|_2^2 \right\}.$$

We proceed by discussing two cases of whether $\bar{x}$ is in the feasible region.

Case 1: $\bar{x} \in \tilde{S}_k(x_t)$**.** Then by the definition of $\bar{x}$, $x_{t+1} = \bar{x}$ and the optimal solution $\bar{x}$ can be computed by setting the derivative of the objective to zero, which yields:

$$x_{t+1} = x_t + \frac{\rho_t}{2} q_t / g_t = x_t + \bar{\rho}_t q_t / g_t,$$

where we can absorb the constant factor into ρ_t. Substitute this solution back to the feasible region and use the fact that $x_{t+1} \in \tilde{S}_k(x_t)$ we get:

$$\rho_t \frac{\langle q_t, \nabla \mathcal{L}(x_t) \rangle}{g_t} + \mathcal{L}(x_t) - \mathcal{L}(w_k) \leq 0. \tag{6}$$

Case 2: $\bar{x} \notin \tilde{S}_k(x_t)$**.** Then the solution x_{t+1} must lie on the boundary of the linearized constraint and can be computed by projecting $\bar{x}$ onto the boundary:

$$H_k(x_t) = \{x \colon \mathcal{L}(x_t) + \nabla \mathcal{L}(x_t)^{\mathrm{T}} (x - x_t) = \mathcal{L}(w_k)\}.$$

The Lagrangian of this problem is:

$$L(x, \lambda) = \frac{1}{2} \|x - \bar{x}\|_2^2 + \lambda \big(\mathcal{L}(x_t) + \langle \nabla \mathcal{L}(x_t), x - x_t \rangle - f(w_k) \big).$$

By KKT condition we have:

$$x_{t+1} = \bar{x} - \lambda \nabla f(x_t).$$

Substituting this back to the Lagrangian we get the dual problem:

$$d(\lambda) = -\frac{1}{2} \lambda^2 g_t + \lambda \big(\mathcal{L}(x_t) + \langle \nabla \mathcal{L}(x_t), \bar{x} - x_t \rangle - f(w_k) \big).$$

Maximizing over this quadratic function gives:

$$\lambda^* = \frac{\mathcal{L}(x_t) - f(w_k) + \rho \langle q_t, \nabla \mathcal{L}(x_t) \rangle / g_t}{g_t}.$$

As a result, we get:

$$x_{t+1} = x_t + \rho_t q_t / g_t - \frac{\mathcal{L}(x_t) - f(w_k) + \rho \langle q_t, \nabla \mathcal{L}(x_t) \rangle / g_t}{g_t} \nabla \mathcal{L}(x_t).$$

Combining the results from two cases we can recover the proposed solution:

$$x_{t+1} = x_t + \rho_t q_t / g_t - \left(\frac{\mathcal{L}(x_t) - f(w_k) + \rho \langle q_t, \nabla \mathcal{L}(x_t) \rangle / g_t}{g_t} \right)_+ \nabla \mathcal{L}(x_t).$$

□

B.2 Proof of Lemma 2.3

Proof. Let $d_t = x_{t+1} - x_t$. Define $\Delta_t(\alpha) = \phi_\gamma(x_t + \alpha d_t) - \phi_\gamma(x_t)$. By applying first-order Taylor's theorem twice, we get

$$\begin{aligned}
\Delta_t(\alpha) &= -\frac{1}{2}\|\nabla\mathcal{L}(x_t + \alpha d_t)\|_2^2 + \gamma\big(\mathcal{L}(x_t + \alpha d_t) - \mathcal{L}(w_k)\big)_+ \\
&\quad + \frac{1}{2}\|\nabla\mathcal{L}(x_t)\|_2^2 - \gamma\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+ \\
&= -\alpha\left\langle\nabla^2\mathcal{L}(x)\nabla\mathcal{L}(x_t), d_t\right\rangle + \gamma\big(\mathcal{L}(x_t) + \alpha\langle\nabla\mathcal{L}(x_t), d_t\rangle - \mathcal{L}(w_k)\big)_+ \\
&\quad - \gamma\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+ + O(\alpha^2) \qquad (7) \\
&\le -\alpha\left\langle\nabla^2\mathcal{L}(x)\nabla\mathcal{L}(x_t), d_t\right\rangle + \gamma(1-\alpha)\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+ \\
&\quad - \gamma\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+ + O(\alpha^2) \qquad (8) \\
&\le -\alpha\left\langle\nabla^2\mathcal{L}(x)\nabla\mathcal{L}(x_t), d_t\right\rangle - \gamma\alpha\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+ + O(\alpha^2),
\end{aligned}$$

where (7) follows Taylor's theorem on $\nabla\mathcal{L}(x_t + \alpha d_t)$ and (8) follows the fact that $\langle\nabla\mathcal{L}(x_t), d_t\rangle \le f(w_k) - f(x_t)$. Dividing both sides by α and setting $\alpha \to 0$ gives:

$$\begin{aligned}
D_\phi(x_t, d_t) &\le -\left\langle\nabla^2\mathcal{L}(x)\nabla\mathcal{L}(x_t), d_t\right\rangle - \gamma\left(\mathcal{L}(x_t) - \mathcal{L}(w_k)\right)_+ \\
&= -\big(\langle q_t, v_t\rangle + \rho_t\|q_t\|_2^2\big) - \gamma\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+
\end{aligned}$$

where we substitute $q_t = \nabla^2\mathcal{L}(x)\nabla\mathcal{L}(x_t)$ to simplify the notation. We also corrected the sign of v_t from the original proof [4]. As a result, since $\rho_t > 0$, if $\mathcal{L}(x_t) - \mathcal{L}(w_k) > 0$, it's sufficient to set

$$-\langle q_t, v_t\rangle - \gamma\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+ \le 0$$

i.e., $\gamma_t \ge -\frac{\langle q_t, v_t\rangle}{g_t(\mathcal{L}(x_t) - \mathcal{L}(w_k))}$, which guarantees d_t is a descent direction. On the other hand, when $\mathcal{L}(x_t) - \mathcal{L}(w_k) \le 0$, $\gamma = 0$, then we need to have $\langle q_t, v_t\rangle + \rho_t\|q_t\|_2^2 \ge 0$, i.e., $\rho_t \ge -\frac{\langle q_t, v_t\rangle}{\|q_t\|_2^2}$. □

B.3 Proof of Proposition 2.4

Proof. The first part directly follows the conclusion of Lemma 2.3. Substituting the upper bound of D_ϕ into the line-search condition, we get:

$$\begin{aligned}
\phi_\gamma(x_{t+1}) &\le -\frac{1}{2}g_t + \gamma_t\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+ - \frac{\langle q_t, v_t\rangle + \rho_t\|q_t\|_2^2}{g_t} - \gamma\big(\mathcal{L}(x_t) - \mathcal{L}(w_k)\big)_+ \\
&\le -\frac{1}{2}g_t - \frac{\langle q_t, v_t\rangle + \rho_t\|q_t\|_2^2}{g_t}.
\end{aligned}$$

□

B.4 Algorithm of Level-Set Teleportation

Algorithm 3 Level-set teleportation

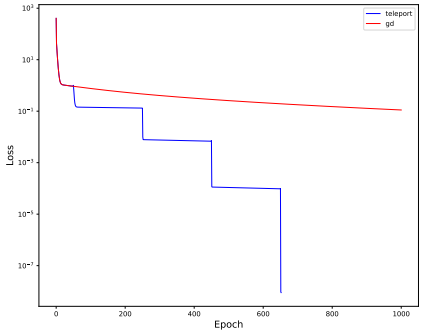
Require: Parameters $\boldsymbol{w}_k$, the loss function $\mathcal{L}$, initial step-size ρ, teleportation steps n_{tel}, line-search steps n_{line}, tolerances ϵ, δ

Ensure: The teleported parameters x_t that have steeper gradients

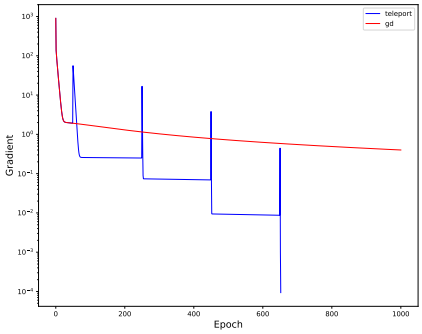
$x_1 \leftarrow w_k$
for $t = 1, \ldots, n_{\text{tel}}$ **do**
  $\rho_t \leftarrow \rho$
  if $\mathcal{L}(x_t) \leq \mathcal{L}(w_k)$ **then**
    $\gamma_t \leftarrow 10^9$
  else
    $\gamma_t \leftarrow -2\frac{\langle q_t, v_t \rangle}{g_t(\mathcal{L}(x_t) - \mathcal{L}(w_k))}$
  end if
  $q_t \leftarrow \nabla^2 \mathcal{L}(x_t) \nabla \mathcal{L}(x_t)$
  $g_t \leftarrow \|\nabla \mathcal{L}(x_t)\|_2^2$
  $v_t \leftarrow -(\rho_t \langle q_t, \nabla \mathcal{L}(x_t) \rangle / g_t + \mathcal{L}(x_t) - \mathcal{L}(w_k))_+ \nabla \mathcal{L}(x_t)$
  $x_{t+1} \leftarrow x_t + (\rho_t \cdot q_t + v_t)/g_t$
  for $i = 1, \ldots, n_{\text{line}}$ **do**
    if $\mathcal{L}(x_t) \leq \mathcal{L}(w_k)$ and $\phi_{\gamma_t}(x_{t+1}) < \phi_{\gamma_t}(x_t)$ **then**
      break
    end if
    if $\mathcal{L}(x_t) > \mathcal{L}(w_k)$ and $\phi_{\gamma_t}(x_{t+1}) \leq \frac{1}{2} g_t - (\langle q_t, v_t \rangle + \rho_t \|q_t\|_2^2)/g_t$ **then**
      break
    end if
    $\rho_t \leftarrow \rho_t / 2$
    $v_t \leftarrow -(\rho_t \langle q_t, \nabla \mathcal{L}(x_t) \rangle / g_t + \mathcal{L}(x_t) - \mathcal{L}(w_k))_+ \nabla \mathcal{L}(x_t)$
    $x_{t+1} \leftarrow x_t + (\rho_t \cdot q_t + v_t)/g_t$
  end for
  $t \leftarrow t + 1$
  if $\|\boldsymbol{P}_t q_t\|_2 \leq \epsilon$ and $\mathcal{L}(x_t) - \mathcal{L}(w_k) \leq \delta$ **then**
    break
  end if
end for
return The teleported parameters $\boldsymbol{w}$

B.5 Addditional Results on Level-Set Teleportation

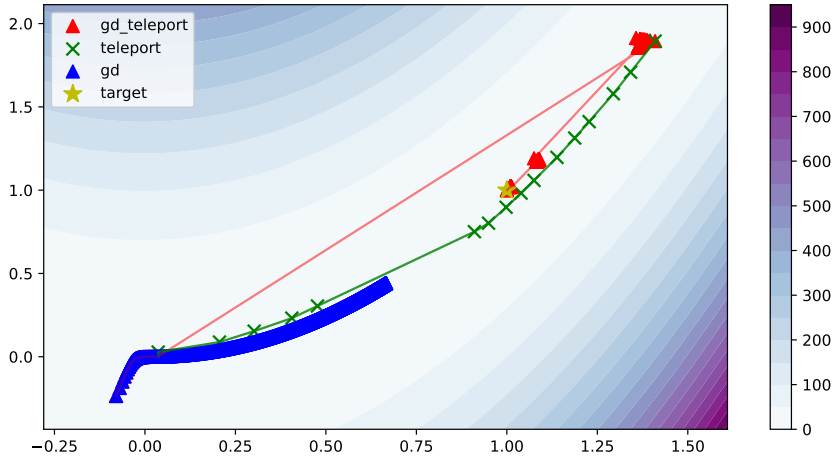
As a complement, we also conduct experiments on test functions using level-set teleportation without line-search on gradient descent. The results are listed in Figure 9, which are consistent to the results with line search. Moreover, without line search, gradient descent is easier to stay in the plateau, resulting in inferior results compared to the one with level-set teleportation.



(a) Loss curve.



(b) Gradient norm curve.



(c) Parameter trajectory.

Figure 9: The loss and gradient norm curves of optimizing the Rosenbrock function with and without level-set teleportation.