
Supplementary Materials for the submission You Shall not Pass: the Zero-Gradient Problem in Predict and Optimize for Convex Optimization

Anonymous Author(s)

Affiliation

Address

email

1 A Proofs

2 *Proof of Lemma 3.4.* Let $\Delta\hat{w}$ denote an arbitrary direction and let $d = \nabla_{\hat{w}} x^*(\hat{w}) \Delta\hat{w}$ be the cor-
3 responding directional derivative of the decision. The existence of d is guaranteed by the strict
4 complementary slackness conditions and Lemma 3.3. Let $t \rightarrow 0^+$. Then, we have

$$\hat{x}'(t) := x^*(\hat{w} + t\Delta\hat{w}) = \hat{x} + td + o_x(t),$$

5 where $o_x(t)$ is the “little o ” notation, i.e., $\lim_{t \rightarrow 0^+} \frac{\|o_x(t)\|_2}{t} = 0$. To prove the lemma, we first want
6 to show that $d^\top n_i = 0, \forall i \in I(\hat{x})$. Then, we will show that it implies the lemma’s claim.

7 By definition, $n_i = \nabla_x g_i(\hat{x})$. Then, since $g_i(\cdot)$ is differentiable and $g_i(\hat{x}) = 0, \forall i \in I(\hat{x})$, we have
8 the following first-order approximation for $g_i(\hat{x}'(t))$:

$$g_i(\hat{x}'(t)) = g_i(\hat{x} + td + o(t)) = g_i(\hat{x}) + tn_i^\top d + o_g(t) = tn_i^\top d + o_g(t).$$

9 Since \hat{x}' is the solution of the internal optimization problem, the inequality $g_i(\hat{x}'(t)) \leq 0$ holds.
10 Hence, the equation above implies that $n_i^\top d \leq 0$. Now, we want to show that, in fact, $n_i^\top d = 0$. For
11 a proof by contradiction, suppose that $n_i^\top d < 0$. Then, by definition of $o_g(t)$, there exists $\epsilon > 0$, such
12 that

$$0 < t < \epsilon \implies g_i(\hat{x}'(t)) < 0.$$

13 Now, we will to show that $g_i(\hat{x}'(t)) < 0$ contradicts the complementary slackness condition at \hat{x} .
14 From Lemma 3.3, we know the KKT multiplier, $\alpha'_i(t) := \alpha_i(\hat{w} + t\Delta\hat{w})$, is a continuous function
15 of t . On the one hand, from the KKT conditions, we know that $g_i(\hat{x}'(t)) < 0 \implies \alpha'_i(t) = 0$.
16 Therefore, $\alpha'_i(t) = 0$ for $t < \epsilon$. Hence, we have

$$\lim_{t \rightarrow 0^+} \alpha'_i(t) = 0.$$

17 On the other hand, the continuity implies that $\lim_{t \rightarrow 0^+} \alpha'_i(t) = \alpha'_i(0) = \alpha_i$ and, due to strict
18 complementary slackness, $\alpha_i > 0$. Hence, we also have

$$\lim_{t \rightarrow 0^+} \alpha'_i(t) > 0.$$

19 We arrived at a contradiction and therefore can claim that $d^\top n_i = 0$ for all n_i . Since $\{n_i | i \in I(\hat{x})\}$
20 is a basis of $\mathcal{N}(\hat{x})$, this implies that for any direction $v \in \mathcal{N}(\hat{x})$ and for any $\Delta\hat{w}$, we have
21 $v^\top \nabla_{\hat{w}} x^*(\hat{w}) \Delta\hat{w} = 0$. In other words, vector $v^\top \nabla_{\hat{w}} x^*(\hat{w})$ is orthogonal to the whole space
22 of \hat{w} and hence it must be zero, $v^\top \nabla_{\hat{w}} x^*(\hat{w}) = 0, \forall v \in \mathcal{N}(\hat{x})$. Hence $\mathcal{N}(\hat{x})$ is contained in the left
23 null space of $\nabla_{\hat{w}} x^*(\hat{w})$. \square

24 *Proof of Lemma 3.6.* First, consider the case when the unconstrained maximum \hat{w} is in the interior
 25 of \mathcal{C} . By definition of x_{QP}^* , it means that $\hat{x} = x_{QP}^*(\hat{w})$ is also in the interior of \mathcal{C} and $\hat{x} = \hat{w}$. Then,
 26 x_{QP}^* is the identity function around \hat{w} , and hence $x_{QP}^*(\hat{w} + \Delta\hat{w}) = x(\hat{w}) + \Delta\hat{w}$ for small enough
 27 $\Delta\hat{w}$. Hence, $\nabla_{\hat{w}} x_{QP}^*(\hat{w}) = I$. Since no constraints are active in this case ($I(\hat{x}) = \emptyset$), the lemma's
 28 claim holds.

29 Now, consider the case when some constraints are active, and thus \hat{x} lies on the boundary of \mathcal{C} . To get
 30 the exact form of the Jacobian $\nabla_x x_{QP}^*(\hat{w})$, we will compute $\lim_{t \rightarrow 0} x_{QP}^*(\hat{w} + t\Delta\hat{w})$ for all possible
 31 $\Delta\hat{w}$. As in the QP case the predictions \hat{w} lie in the same space as \hat{x} , we can do it first for $\Delta\hat{w} \in \mathcal{N}(\hat{x})$
 32 and then for $\Delta\hat{w} \perp \mathcal{N}(\hat{x})$.

33 **1.** $\Delta\hat{w} \in \mathcal{N}(\hat{x})$. For $\Delta\hat{w} \in \mathcal{N}(\hat{x})$, we want to show that the corresponding directional derivative is
 34 zero. We begin by computing the internal gradient $\nabla_x f_{QP}(\hat{x}, \hat{w})$:

$$\nabla_x f_{QP}(\hat{x}, \hat{w}) = -\nabla_x \|x - w\|_2^2 = 2(\hat{w} - \hat{x}).$$

35 Using this formula, we can write the internal gradient for the perturbed prediction $\hat{w} + t\Delta\hat{w}$ at the
 36 same point \hat{x} :

$$\nabla_x f_{QP}(\hat{x}, \hat{w} + t\Delta\hat{w}) = \nabla_x f_{QP}(\hat{x}, \hat{w}) + 2t\Delta\hat{w}.$$

37 By definition, $\mathcal{N}(\hat{x})$ is a linear span of the vectors $\{n_i | i \in I(\hat{x})\}$. Hence, since $\Delta\hat{w} \in \mathcal{N}(\hat{x})$, it can
 38 be expressed as

$$\Delta\hat{w} = \sum_{i \in I(\hat{x})} \delta_i n_i, \quad \delta_i \in \mathbb{R}. \quad (*)$$

39 By Property 3.2, the internal gradient has the following representation:

$$\nabla_x f_{QP}(\hat{x}, \hat{w}) = \sum_{i \in I(\hat{x})} \alpha_i n_i, \quad \alpha_i > 0. \quad (**)$$

40 Then, combining (*) and (**), we obtain

$$\nabla_x f_{QP}(\hat{x}, \hat{w} + t\Delta\hat{w}) = \nabla_x f_{QP}(\hat{x}, \hat{w}) + 2t\Delta\hat{w} = \sum_{i \in I(\hat{x})} (\alpha_i + 2t\delta_i) n_i$$

41 Since $\alpha_i > 0, \forall i \in I(\hat{x})$, there exists $\epsilon > 0$, such that $\alpha_i - 2t\delta_i > 0$ for $|t| < \epsilon$. Therefore,
 42 $\nabla_x f_{QP}(\hat{x}, \hat{w} + t\Delta\hat{w})$ lies in the gradient cone of \hat{x} , and hence, by Property 3.2, $x_{QP}^*(\hat{w} + t\Delta\hat{w}) = \hat{x}$
 43 for $|t| < \epsilon$. Therefore, the directional derivative of $x_{QP}^*(\hat{w})$ along $\Delta\hat{w} \in \mathcal{N}(\hat{x})$ is zero.

44 **2.** $\Delta\hat{w} \perp \mathcal{N}(\hat{x})$. Next, let $\Delta\hat{w}$ be orthogonal to $\mathcal{N}(\hat{x})$. We begin with the first order approximation
 45 of $\hat{x}'(t)$:

$$\hat{x}'(t) = \hat{x} + td + o(t).$$

46 From the proof of Lemma 3.3, we can know that $d \perp \mathcal{N}$. By definition of x_{QP}^* , we know that \hat{x} is
 47 the point on \mathcal{C} closest to \hat{w} . Likewise, $\hat{x}'(t)$ is the point on \mathcal{C} closest to $\hat{w} + t\Delta\hat{w}$. Hence, $d = \Delta\hat{w}$.
 48 Therefore, for any $\Delta\hat{w} \perp \mathcal{N}$, the directional derivative of $x_{QP}^*(\hat{w})$ along $\Delta\hat{w}$ is one.

49 So, we have shown that

$$\nabla_{\hat{w}} x_{QP}^*(\hat{w}) \Delta\hat{w} = \begin{cases} 0 & \text{for } \Delta\hat{w} \in \mathcal{N}(\hat{x}) \\ \Delta\hat{w} & \text{for } \Delta\hat{w} \perp \mathcal{N}(\hat{x}). \end{cases}$$

50 Therefore, the lemma is proven. \square

51 *Proof of Theorem 3.9.* First, we want to construct an orthogonal basis $\{e_1, \dots, e_n\}$ of \mathbb{R}^n that will
 52 greatly simplify the calculations. We start by including the internal gradient in this basis, i.e., we define
 53 $e_1 = \nabla_x f_{QP}(\hat{x}, \hat{w})$. Then, let $I(\hat{x}) = \{i | g_i(\hat{x}) = 0\}$ be the set of indices of the active constraints
 54 of the original problem and let $\mathcal{N}(\hat{x}) = \text{span}(\{n_i | i \in I(\hat{x})\})$ be a linear span of their normals. By
 55 the liner independence condition from Assumption 2, $\dim(\mathcal{N}(\hat{x})) = |I(\hat{x})|$. Moreover, by Property
 56 3.2, we know that $e_1 \in \mathcal{N}(\hat{x})$. Then, we can choose vectors $e_2, \dots, e_{|I(\hat{x})|}$ that complement e_1 to
 57 an orthogonal basis of $\mathcal{N}(\hat{x})$. The remaining vectors $e_{|I(\hat{x})|+1}, \dots, e_n$, are chosen to complement
 58 $e_1, \dots, e_{|I(\hat{x})|}$ to an orthogonal basis of \mathbb{R}^n . The choice of this basis is motivated by Lemma 3.6:

59 e_1 is a basis of the null-space of the r -smoothed Jacobian, $e_1, \dots, e_{|I(\hat{x})|}$ form a basis of the null
60 space of the true QP Jacobian, and the remaining vectors form a basis of space in which we can move
61 $x_{QP}^*(\hat{w})$.

62 For brevity, let $f_x = \nabla_x f(\hat{x}, w)$ denote the true gradient vector. By definition, $\Delta\hat{w} = f_x \nabla_{\hat{w}} x_r^*(\hat{x}, \hat{w})$
63 is obtained via the r -smoothed problem. From Property 3.8, we know that $\Delta\hat{w}$ is a projection of f_x
64 on the vectors e_2, \dots, e_n . Then, since e_1, \dots, e_n is an orthogonal basis, we have

$$\Delta\hat{w} = \sum_{i=2}^n \beta_i e_i, \quad \beta_i = f_x^\top e_i, \quad i = 2, \dots, n.$$

65 Now, let's see how this $\Delta\hat{w}$ affects the true decision $x_{QP}^*(\hat{w} + t\Delta\hat{w})$ for $t \rightarrow 0^+$. First, we have a
66 first-order approximation

$$x_{QP}^*(\hat{w} + t\Delta\hat{w}) = \hat{x} + td + o(t),$$

67 for some $d \in \mathbb{R}$. From Lemma 3.6, we know that d is actually a projection of $\Delta\hat{w}$ onto the vectors
68 $e_{|I(\hat{x})|+1}, \dots, e_n$. Therefore, we have

$$x_{QP}^*(\hat{w} + t\Delta\hat{w}) = \hat{x} + \sum_{i=|I(\hat{x})|+1}^n \beta_i e_i + o(t).$$

69 Finally, the change in the true objective can be expressed as

$$\begin{aligned} f(x_{QP}^*(\hat{w} + t\Delta\hat{w}), w) - f(x_{QP}^*(\hat{w}), w) &= t f_x^\top \left(\sum_{i=|I(\hat{x})|+1}^n \beta_i e_i \right) + o(t) = \\ &= t \sum_{i=|I(\hat{x})|+1}^n \beta_i f_x^\top e_i + o(t) = t \sum_{i=|I(\hat{x})|+1}^n \beta_i^2 + o(t) \geq 0. \end{aligned}$$

70 Therefore, perturbing prediction along $\Delta\hat{w}$ does not decrease the true objective $f(\hat{x}, w)$, and hence

$$f(x_{QP}^*(\hat{w} + t\Delta\hat{w}), w) \geq f(x_{QP}^*(\hat{w}), w)$$

71 for $t \rightarrow 0^+$. □

72 B Equality constraints

73 Assumption 2 postulates that for any $x \in \mathcal{C}$, the gradients of active constraints, $\{\nabla_x g_i(x) | g_i(x) = 0\}$,
74 are linearly independent. Now, suppose we include equality constraints in our problem. e.g., we have
75 a constraint $g^{eq}(x) \leq 0$ and $-g^{eq}(x) \leq 0$ for some g . Clearly, the gradients of $g^{eq}(x)$ and $-g^{eq}(x)$
76 violate the independence assumption. However, we claim that it does not affect our results. Let \hat{w}
77 and \hat{x} be a prediction and a corresponding decision and let $n^{eq} = \nabla_x g^{eq}(\hat{x})$. Suppose the equality
78 constraint $g^{eq}(\hat{x}) = 0$ is active. Let $I(\hat{x})$ be the set of indices of the active constraints *not including*
79 $g^{eq}(x)$. Then, we have a representation of the internal gradient,

$$\nabla_x f(\hat{x}, \hat{w}) = \alpha_1^{eq} n^{eq} - \alpha_2^{eq} n^{eq} + \sum_{i \in I(\hat{x})} \alpha_i n_i.$$

80 Suppose that $\alpha_1^{eq} \neq \alpha_2^{eq}$, e.g., without loss of generality, $\alpha_1^{eq} > \alpha_2^{eq}$. Then,

$$\nabla_x f(\hat{x}, \hat{w}) = (\alpha_1^{eq} - \alpha_2^{eq}) n^{eq} + \sum_{i \in I(\hat{x})} \alpha_i n_i$$

81 and hence removing the constraint $-g^{eq}(x) \leq 0$ would not change the optimality of \hat{x} . The remaining
82 problem would satisfy complementary slackness and hence would have all the properties demonstrated
83 in Section 3. Therefore, for the case with equality constraints, we need to extend the complementary
84 slackness conditions by demanding $\alpha_1^{eq} \neq \alpha_2^{eq}$.

	Search space
<i>Learning rate</i>	$\{5 \times 10^{-6}, 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}\}$
<i>Batch size</i>	$\{1, 2, 4, 8, 32\}$
<i>Proj. distance weight α from Eq. (6)</i>	$\{0.001, 0.0025, 0.005, 0.01, 0.05, 0.1, 1\}$
x_{shift}	$\{0, .1, 1\}$
x_{scale}	$\{0.1, 1, 5\}$

Table 1: Search space for different hyperparameters for the portfolio optimization problem

	$\lambda = 2$	$\lambda = 1$	$\lambda = 0.5$	$\lambda = 0.25$	$\lambda = 0.1$	$\lambda = 0$
<i>Learning rate</i>	10^{-5}	10^{-5}	2×10^{-5}	2×10^{-5}	2×10^{-5}	5×10^{-5}
<i>Batch size</i>	1	1	1	1	1	1
<i>Penalty weight α from Eq. (6)</i>	0.1	0.1	0.02	0.005	0.005	0.0025
<i>Training epochs</i>	180	180	180	180	180	180
x_{shift}	1	1	1	1	1	1
x_{scale}	0.1	0.1	0.1	0.1	0.1	0.1

Table 2: Best performing values of the hyperparameters for the portfolio optimization problem with different λ 's

85 C Experimental details

86 In this section, we provide the details of the experiments reported in the paper. All experiments
87 were conducted on a machine with 32gb RAM and NVIDIA GeForce RTX 3070. The code is
88 written in Python 3.8, and neural networks are implemented in *PyTorch* 1.12. For methods requiring
89 differentiation of optimization problems (those, without r -smoothing), we use the implementation
90 by Agrawal et. al [2019a]. The code can be found at *placeholder for GitHub link*. For the reviewers,
91 the code is submitted through *OpenReview*.

92 C.1 Portfolio optimization problem

93 In the portfolio optimization problem, the predictor ϕ_θ is represented by a fully connected neural
94 network with two hidden layers of 100 neurons each, and *ReLU* activation functions. The output layer
95 has no activation function. Instead, the output of the neural network is scaled by the factor x_{scale}
96 and shifted by x_{shift} . For the methods using the QP approximation, the output layer predicts only \hat{w}
97 and consists of 200 neurons, one per the decision variable x_i . For the method that uses the original
98 problem formulation and predicts both \hat{p} and \hat{Q} , the output layer additionally predicts a 200×200
99 matrix L and then sets $\hat{Q} := (0.9L + 0.1I)(0.9L + 0.1I)^\top$, where I is the identity matrix.

100 For training, we used the *Adam* optimizer from *PyTorch*, with custom learning rate and otherwise
101 default parameters. The values of different hyperparameters were determined by a grid search
102 procedure, summarized in Table 1. The values used in the experiments are reported in Table 2. These
103 values may vary across experiments with different λ 's. For each λ , however, the four studied methods
104 use the same values of the hyperparameters (the only exception is the projection distance weight α ,
105 which is always zero for methods without regularization).

106 C.2 Optimal power flow problem

	Search space
<i>Learning rate</i>	$\{5 \times 10^{-6}, 10^{-5}, 2 \times 10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}\}$
<i>Batch size</i>	$\{1, 2, 4, 8, 32\}$
<i>Proj. distance weight α from Eq. (6)</i>	$\{0.0001, 0.00025, 0.0005, 0.001, 0.005, 0.01, 0.1\}$
x_{shift}	$\{0, .1, 1\}$
x_{scale}	$\{0.1, 1, 5\}$

Table 3: Search space for different hyperparameters in the DC OPF problem

	Value
<i>Learning rate</i>	5×10^{-5}
<i>Batch size</i>	1
<i>Penalty weight α from Eq. (6)</i>	0.0001
<i>Training epochs</i>	250
x_{shift}	7
x_{scale}	1

Table 4: Best performing values of the hyperparameters for the DC OPF problem

107 **Data generation process.** Data for the DC OPF problem is generated artificially. First, we
108 randomly generate a grid topology, see Figure 1 for an example. For each line, its admittance
109 is set to $6S$. Nodal voltages are bounded between 325V and 375V, and the reference node has
110 a fixed voltage of $v_0 = 350V$. The demand in loads (power upper-bound), generators capacity
111 (power lower-bound), and line current limits are sampled randomly from the following normal
112 distributions: $\mathcal{N}(8000, 2500) \times \text{watt-hour}$, $\mathcal{N}(-14000, 2500) \times \text{watt-hour}$, $\mathcal{N}(25, 5) \times \text{ampere}$. The
113 coefficients w are also sampled from the normal distributions: $\mathcal{N}(1.2, 1)$ for loads, and $\mathcal{N}(0.8, 0.1)$
114 for generators. Finally, all values are normalized such that v_0 becomes 7V (surprisingly, it performed
115 better numerically than scaling v_0 to 1V). The observations o consist of the true coefficients w ,
116 demand of the loads, the capacity of the generators, and line current limits.

117 The predictor in the optimal power flow problem is the same as the one in the portfolio optimization,
118 except for its hidden layers consisting of 256 neurons and using *LeakyReLU* activation functions.
119 The hyperparameters search space and final values are reported in Tables 3,4

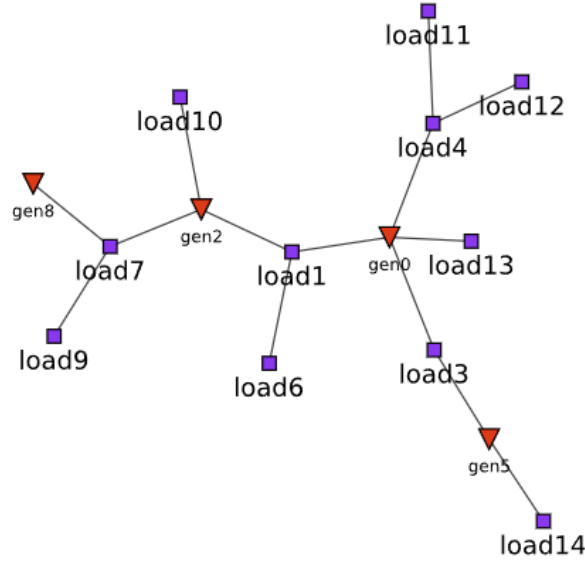


Figure 1: Example of randomly generated grid topology. Red triangles represent generator nodes, and purple squares represent loads.