We will present our theorem of transforming a constrained linear programming problem into an unconstrained solving problem amenable to GPU acceleration. Preliminaries and notations of the problem statement are provided as follows.

Preliminaries. Given a network with $L+1$ layers in total and each layer corresponds to a layer index, the *input layer* is at index 0 and the output layer is at index L. We denote a set $[R]$ that includes all ReLU layer indexes; the set L_1 contains all linear layer indexes with one connected preceding layer. The set L_2 records all indexes of linear layers that take two preceding layers. We assume that $[R] \cup [L_1] \cup [L_2] = [1, \dots, L]$ and both $1, L \in [L_1]$.

The output of a ReLU layer is represented by $\hat{x}^{(i)}, i \in [R]$; $\hat{x}_p^{(i)}, i \in [R]$ as the input/preceding layer of the ReLU layer. Given the neuron index j and layer index i, $\hat{x}^{(i)j}$ represents the j-th neuron at i-th layer and $\hat{x}_p^{(i)j}$ refers to its input neuron. $x^{(i)}, i \in [L_1] \cup [L_2]$ represents the output of a linear layer; $\hat{x}^{(0)}, x^{(0)}$ both denotes the input layer. Symbol $x_p^{(i)}, i \in [L_1]$ refers to the predecessor of layer $x^{(i)}$ for $i \in [L_1]$; whereas $x_{p_1}^{(i)}, x_{p_2}^{(i)}$ are the two preceding layers of layer $x^{(i)}$ for $i \in [L_2]$. Finally, we designate $S(i)$ as a set that includes the indexes of all connected succeeding layers of layer i and $i_s \in S(i)$; the set $S^2(i) = \cup_{i_s \in S(i)} S(i_s)$, which includes the successors' indexes of succeeding layers of layer i and $i_{s^2} \in S^2(i)$.

Theorem 1. Given the original constrained problem formulation as follows:

$$\begin{aligned}
&\min_{x,\hat{x}} c^{(0)} \hat{x}^{(0)} + \sum_{i \in [R]} c^{(i)T} \hat{x}_p^{(i)} \\
&\text{s.t. } l^{(0)} \le \hat{x}^{(0)} \le u^{(0)}; Hx^{(L)} + d \le 0 \\
&x^{(i)} = W^{(i)} x_p^{(i)} + b^{(i)}, \text{for } i \in [L_1] \\
&x^{(i)} = x_{p_1}^{(i)} + x_{p_2}^{(i)}, \text{for } i \in [L_2] \\
&\hat{x}^{(i)j} = \hat{x}_p^{(i)j}, \text{for } i \in [R], j \in I^{+(i)} \\
&\hat{x}^{(i)j} = 0, \text{for } i \in [R], j \in I^{-(i)} \\
&\hat{x}^{(i)j} \ge 0, \hat{x}^{(i)j} \ge \hat{x}_p^{(i)j}, \text{for } i \in [R], j \in I^{\pm(i)} \\
&\hat{x}^{(i)j} \le \frac{u^{(i)j}}{u^{(i)j} - l^{(i)j}} (\hat{x}_p^{(i)j} - l^{(i)j}), \text{for } i \in [R], j \in I^{\pm(i)} \\
&P^{(i)} \hat{x}_p^{(i)} + \hat{P}^{(i)} \hat{x}^{(i)} - p^{(i)} \le 0, \text{for } i \in [R]
\end{aligned} \tag{1}$$

In detail, $l^{(0)}, u^{(0)}$ record the lower and upper bounds of input neurons; $Hx^{(L)} + d \le 0$ represents the output constraints that encode the existence of multiple adversarial examples. For ReLU neurons, their functionalities depend on the stability statuses. For example, suppose a linear layer i is followed by a ReLU layer i_s. We define a ReLU neuron to be stably activated if it takes non-negative input interval at layer i and therefore, it is equivalent to its input neuron, and we collect the indexes of those non-negative input neurons at layer i as $I^{+(i)}$ and those stably activated neurons at layer i_s as $I^{+(i_s)}$. On the other hand, for the stably deactivated ReLU neurons that only take non-positive input intervals, their outputs are always evaluated to 0, and we denote the indexes of those non-positive input neurons/stably deactivated ReLU neurons as a set $I^{-(i)}/I^{-(i_s)}$. The indexes of those linear neurons that take both positive and negative values are recorded in $I^{\pm(i)}$, and their succeeding ReLU neurons are unstable and recorded in $I^{\pm(i_s)}$. In particular, the unstable ReLU neuron is a non-linear function, and we use an orange-colored triangle shape (defined by the three linear constraints) to over-approximate its behavior as Figure 1 illustrates, where $l^{(i)j}, u^{(i)j}$ record its lower input bound and upper input bound. For simplicity, we denote $\frac{u^{(i)j}}{u^{(i)j} - l^{(i)j}}$ as $s^{(i)j}$. Constraints $P^{(i)} \hat{x}_p^{(i)} + \hat{P}^{(i)} \hat{x}^{(i)} - p^{(i)} \le 0$ are the multi-ReLU constraints that capture the dependencies of multiple ReLU neurons in the same layer from WraLU method. The coefficients $c^{(0)}$ and $c^{(i)}, i \in [R]$ are used to control the objective function of our problem. As we aim to resolve the input lower and upper bounds of unstable ReLU neurons, we only set one element among $c^{(0)}, c^{(i)}, i \in [R]$ as 1 (for lower bound computation) or -1 (for upper bound) for the respective neuron, the rest of the elements are set as 0.

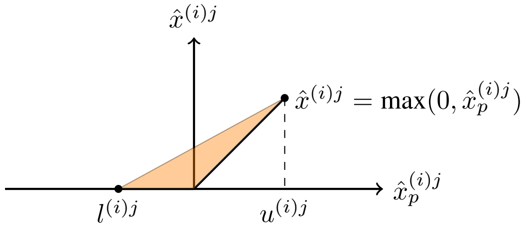

Figure 1: The approximation of a ReLU neuron

Eventually, we transform the constrained solving problem stated in Equation 1 to an unconstrained problem as shown below, where we annotate $[x]_+ = \max(x, 0), [x]_- = -\min(x, 0)$:

$$\max_{\gamma,v,\pi,\alpha} [c^{(0)T} - v^{(1)T} w^{(1)}]_+ \cdot l^{(0)} - [v^{(1)T} w^{(1)} - c^{(0)T}]_+ \cdot u^{(0)} \tag{2}$$

$$+\gamma^T d + \sum_{i \in [R]} \sum_{j \in I^{\pm(i)}} [\hat{v}^{(i)j}]_+ \cdot s^{(i)j} \cdot l^{(i)j} \tag{3}$$

$$- \sum_{i \in [R]} \pi^{(i)T} p^{(i)} - \sum_{i \in [L_1]} v^{(i)T} b^{(i)} \tag{4}$$

$$\text{s.t. } v^{(L)} = -H^T \gamma;\ \gamma, \pi \ge 0;\ \alpha \in [0, 1] \tag{5}$$

$$\text{for } i \in [L_1] \cup [L_2] \text{ and } i_s \in [R] \cap S(i) \text{ and } i_s \notin [L_2]: \tag{6}$$

$$v^{(i)j} = -c^{(i_s)j}, j \in I^{-(i)} \tag{7}$$

$$v^{(i)j} = \sum_{i_{s^2} \in S(i_s) \cap [L_1]} v^{(i_{s^2})T} W_{:,j}^{(i_{s^2})} - c^{(i_s)j}, j \in I^{+(i)} \tag{8}$$

$$\text{for } j \in I^{\pm(i)}: \tag{9}$$

$$v^{(i)j} = s^{(i_s)j} [\hat{v}^{(i_s)j}]_+ - c^{(i_s)j} - \pi^{(i_s)T} P_{:,j}^{(i_s)} - \alpha^{(i_s)j} [\hat{v}^{(i_s)j}]_- \tag{10}$$

$$\hat{v}^{(i)j} = \sum_{i_{s^2} \in S(i_s) \cap [L_1]} v^{(i_{s^2})T} W_{:,j}^{(i_{s^2})} - \pi^{(i_s)T} \hat{P}_{:,j}^{(i_s)} \tag{11}$$

$$\text{for } i \in [L_1] \text{ and } i_s \in [L_2] \cap S(i) \text{ and } i_s \notin [R]: \tag{12}$$

$$v^{(i)} = v^{(i_s)} \tag{13}$$

Here, any valid setting of $\gamma, \pi \geq 0; \alpha \in [0,1]$ leads to a safe lower bound of the original problem. Based on the values of γ, π, α, we compute the values of $v^{(i)}$ and $\hat{v}^{(i)}$ (if any) in an inverse way from $v^{(L)}$ to $v^{(0)}$. By leveraging all assignments of variables, we could compute the objective value stated at Equation 2 - 4. In practice, the solving process starts with a valid initialization of γ, π, α, then we optimize these variables using gradient information.

Proof. Next, we present our derivation of the Lagrangian dual formulation presented at Equation 2 - 13 from the original problem statement at Equation 1. Firstly, we take the Lagrangian dual of most of the constraints in Equation 1, and we will obtain the following formulation:

$$\min_{x,\hat{x}} \max_{\gamma,v,\mu,\lambda,\pi,\tau} c^{(0)T}\hat{x}^{(0)} + \gamma^T(Hx^{(L)} + d) \tag{14}$$

$$+ \sum_{i\in[R]} c^{(i)T}\hat{x}_p^{(i)} \tag{15}$$

$$+ \sum_{i\in[L_1]} v^{(i)T}(x^{(i)} - W^{(i)}x_p^{(i)} - b^{(i)}) \tag{16}$$

$$+ \sum_{i\in[L_2]} v^{(i)T}(x^{(i)} - x_{p_1}^{(i)} - x_{p_2}^{(i)}) \tag{17}$$

$$+ \sum_{i\in[R]} \sum_{j\in I^{\pm(i)}} [\mu^{(i)j}(-\hat{x}^{(i)j}) + \tau^{(i)j}(\hat{x}_p^{(i)j} - \hat{x}^{(i)j}) \tag{18}$$

$$+\lambda^{(i)j}(\hat{x}^{(i)j} - s^{(i)j}\hat{x}_p^{(i)j} + s^{(i)j}l^{(i)j})] \tag{19}$$

$$+ \sum_{i\in[R]} \sum_{j\in I^{\pm(i)}} (\pi^{(i)T}P_{:,j}^{(i)}\hat{x}_p^{(i)j} + \pi^{(i)T}\hat{P}_{:,j}^{(i)}\hat{x}^{(i)j}) \tag{20}$$

$$- \sum_{i\in[R]} \pi^{(i)T}p^{(i)} \tag{21}$$

$$\text{s.t. } l^{(0)} \leq \hat{x}^{(0)} \leq u^{(0)} \tag{22}$$

$$\hat{x}^{(i)j} = \hat{x}_p^{(i)j}, \text{for } i \in [R], j \in I^{+(i)} \tag{23}$$

$$\hat{x}^{(i)j} = 0, \text{for } i \in [R], j \in I^{-(i)} \tag{24}$$

$$\gamma, \mu, \lambda, \pi, \tau \geq 0 \tag{25}$$

For inequality constraints, we add Lagrangian multipliers $\gamma, \mu, \lambda, \pi, \tau \geq 0$; for equality constraints of linear computation $x^{(i)} = W^{(i)}x_p^{(i)} + b^{(i)}, x^{(i)} = x_{p_1}^{(i)} + x_{p_2}^{(i)}$, the Lagrangian multipliers $v^{(i)T}$ are unbounded.

The constraints of stably activated ReLU neurons at $I^{+(i_s)}$ will be handled by taking the ReLU neuron $\hat{x}^{(i)j}$ and its input linear neuron $\hat{x}_p^{(i)j}$ as the same. Therefore we have term $\sum_{i_s\in S(i)\cap[R]} \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W_{:,j}^{(i_{s2})}$ for $x^{(i)j}, j \in I^{+(i)}$ at later Equation 30, due the fact that this linear neuron $x^{(i)j}$ equals to its ReLU neuron $\hat{x}^{(i_s)j}$ which might be fully-connected to a subsequent linear layer at index i_{s^2} that includes constraint $v^{(i_{s2})T}(x^{(i_{s2})} - W^{(i_{s2})}\hat{x}^{(i_s)j} - b^{(i_{s2})})$. For the stably deactivated ReLU neurons at $I^{-(i_s)}$, they are simply treated as constant 0.

Please note that Equation 21 - 22 are reformulated from $\sum_{i\in[R]} \pi^{(i)T}(P^{(i)}\hat{x}_p^{(i)} + \hat{P}^{(i)}\hat{x}^{(i)} - p^{(i)})$, where $P^{(i)}\hat{x}_p^{(i)} + \hat{P}^{(i)}\hat{x}^{(i)} - p^{(i)}$ comes from the multi-ReLU constraints. As multi-ReLU constraints only handle *unstable* neurons, we explicitly take the coefficients of unstable neurons in $I^{\pm(i)}$ into consideration; whereas the stable neurons get zero coefficients, thus we ignore them. The subscript $:, j$ means the j-th column of the matrix.

In the next step, we switch the max and min operators and reformulate the equation concerning different $x^{(i)}, \hat{x}^{(i)}$:

$$\max_{\gamma,v,\mu,\lambda,\pi,\tau} \min_{x,\hat{x}} (c^{(0)T} - v^{(1)T}W^{(1)})\hat{x}^{(0)} \tag{26}$$

$$+(\gamma^T H + v^{(L)T}) \cdot x^{(L)} \tag{27}$$

$$+ \sum_{i\in[L_1]} \sum_{j\in I^{-(i)}} (v^{(i)j} + \sum_{i_s\in S(i)\cap[R]} c^{(i_s)j} - \sum_{i_s\in S(i)\cap[L_2]} v^{(i_s)j}) \cdot x^{(i)j} \tag{28}$$

$$+ \sum_{i\in[L_1]} \sum_{j\in I^{+(i)}} [v^{(i)j} + \sum_{i_s\in S(i)\cap[R]} c^{(i_s)j} - \sum_{i_s\in S(i)\cap[L_2]} v^{(i_s)j} \tag{29}$$

$$- \sum_{i_s\in S(i)\cap[R]} \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W_{:,j}^{(i_{s2})}] \cdot x^{(i)j} \tag{30}$$

$$+ \sum_{i\in[L_1]} \sum_{j\in I^{\pm(i)}} \{[v^{(i)j} - \sum_{i_s\in S(i)\cap[L_2]} v^{(i_s)j} \tag{31}$$

$$+\sum_{i_s\in S(i)\cap[R]}(c^{(i_s)j} + \tau^{(i_s)j} - \lambda^{(i_s)j}s^{(i_s)j} + \pi^{(i_s)T}P_{:,j}^{(i_s)})] \cdot x^{(i)j} \tag{32}$$

$$+ \sum_{i_s\in S(i)\cap[R]} [\lambda^{(i_s)j} - \mu^{(i_s)j} - \tau^{(i_s)j} + \pi^{(i_s)T}\hat{P}_{:,j}^{(i_s)} \tag{33}$$

$$- \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W_{:,j}^{(i_{s2})}] \cdot \hat{x}^{(i_s)j}\} \tag{34}$$

$$+ \sum_{i\in[L_2]} \sum_{j\in I^{-(i)}} (v^{(i)j} + \sum_{i_s\in S(i)\cap[R]} c^{(i_s)j}) \cdot x^{(i)j} \tag{35}$$

$$+ \sum_{i\in[L_2]} \sum_{j\in I^{+(i)}} [v^{(i)j} + \sum_{i_s\in S(i)\cap[R]} c^{(i_s)j} \tag{36}$$

$$- \sum_{i_s\in S(i)\cap[R]} \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W_{:,j}^{(i_{s2})}] \cdot x^{(i)j} \tag{37}$$

$$+ \sum_{i\in[L_2]} \sum_{j\in I^{\pm(i)}} \{x^{(i)j} \cdot [v^{(i)j} \tag{38}$$

$$+\sum_{i_s\in S(i)\cap[R]}(c^{(i_s)j} + \tau^{(i_s)j} - \lambda^{(i_s)j}s^{(i_s)j} + \pi^{(i_s)T}P_{:,j}^{(i_s)})] \tag{39}$$

$$+ \sum_{i_s\in S(i)\cap[R]} [\lambda^{(i_s)j} - \mu^{(i_s)j} - \tau^{(i_s)j} + \pi^{(i_s)T}\hat{P}_{:,j}^{(i_s)} \tag{40}$$

$$- \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W_{:,j}^{(i_{s2})}] \cdot \hat{x}^{(i_s)j}\} \tag{41}$$

$$+ \sum_{i\in[R]} \sum_{j\in T^{\pm(i)}} \lambda^{(i)j}s^{(i)j}l^{(i)j} - \sum_{i\in[L_1]} v^{(i)T}b^{(i)} \tag{42}$$

$$+\gamma^T d - \sum_{i\in[R]} \pi^{(i)T}p^{(i)} \tag{43}$$

$$\text{s.t. } l^{(0)} \leq \hat{x}^{(0)} \leq u^{(0)} \ ; \ \gamma, \mu, \lambda, \pi, \tau \geq 0 \tag{44}$$

For the equation presented above, line 28 - 34 are terms of L_1 layers and line 35 - 41 are terms of L_2 layers. To simplify the terms, we make an assumption about the network architecture: (1) L_2 layer is always followed by a ReLU layer (otherwise two consecutive linear layers will collapse to just one linear layer); (2) L_1 layer is either followed by one ReLU layer or one L_2 layer, but not both. As such, the formula will be further reconstructed based on the succeeding layer type:

$$\max_{\gamma,v,\mu,\lambda,\pi,\tau} \min_{x,\hat{x}} (c^{(0)T} - v^{(1)T}W^{(1)})\hat{x}^{(0)} \tag{45}$$

$$+(\gamma^T H + v^{(L)T}) \cdot x^{(L)} \tag{46}$$

$$+\sum_{i\in[L_1]} \sum_{i_s\in S(i)\cap[L_2]} (v^{(i)T} - v^{(i_s)T}) \cdot x^{(i)} \tag{47}$$

$$+\sum_{i\in[L_1]\cup[L_2]} \sum_{i_s\in S(i)\cap[R]} \sum_{j\in I^{-(i)}} (v^{(i)j} + c^{(i_s)j}) \cdot x^{(i)j} \tag{48}$$

$$+\sum_{i\in[L_1]\cup[L_2]} \sum_{i_s\in S(i)\cap[R]} \sum_{j\in I^{+(i)}} (v^{(i)j} + c^{(i_s)j} \tag{49}$$

$$-\sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W^{(i_{s2})}_{:,j}) \cdot x^{(i)j} \tag{50}$$

$$+\sum_{i\in[L_1]\cup[L_2]} \sum_{i_s\in S(i)\cap[R]} \sum_{j\in I^{\pm(i)}} [(v^{(i)j} + c^{(i_s)j} + \tau^{(i_s)j} \tag{51}$$

$$-\lambda^{(i_s)j}s^{(i_s)j} + \pi^{(i_s)T}P^{(i_s)}_{:,j}) \cdot x^{(i)j} \tag{52}$$

$$+(\lambda^{(i_s)j} - \mu^{(i_s)j} - \tau^{(i_s)j} + \pi^{(i_s)T}\hat{P}^{(i_s)}_{:,j} \tag{53}$$

$$-\sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W^{(i_{s2})}_{:,j}) \cdot \hat{x}^{(i_s)j}] \tag{54}$$

$$+\sum_{i\in[R]} \sum_{j\in I^{\pm(i)}} \lambda^{(i)j}s^{(i)j}l^{(i)j} - \sum_{i\in[L_1]} v^{(i)T}b^{(i)} \tag{55}$$

$$+\gamma^T d - \sum_{i\in[R]} \pi^{(i)T}p^{(i)} \tag{56}$$

$$\text{s.t. } l^{(0)} \le \hat{x}^{(0)} \le u^{(0)} \;;\; \gamma,\mu,\lambda,\pi,\tau \ge 0 \tag{57}$$

Term 45 is related to $\hat{x}^{(0)}$ and constrained by $l^{(0)} \le \hat{x}^{(0)} \le u^{(0)}$, therefore it can be minimized as:

$$[c^{(0)T} - V^{(1)T}W^{(1)}]_+ l^{(0)} - [V^{(1)T}W^{(1)} - c^{(0)T}]_+ u^{(0)} \tag{58}$$

For other $x^{(i)}$ and $\hat{x}^{(i)}$ whose values are unbounded, let their coefficients be nonzero. Then, the negative coefficients could be combined with $+\infty$ assignments to $x^{(i)}$ and $\hat{x}^{(i)}$ while positive coefficients could be computed with $-\infty$, which leads to the inner minimization with value $-\infty$. Therefore, for the maximization outside to reach optimization, those coefficients must be zero, a contradiction. As such, the coefficients of those unbounded constraints must be 0, which brings forth:

$$\gamma^T H + v^{(L)T} = 0 \tag{59}$$

$$\text{for } i \in [L_1] \text{ and } i_s \in S(i) \cap [L_2] \text{ and } i_s \notin [R] : \tag{60}$$

$$v^{(i)T} - v^{(i_s)T} = 0 \tag{61}$$

$$\text{for } i \in [L_1] \cup [L_2] \text{ and } i_s \in S(i) \cap [R] : \tag{62}$$

$$v^{(i)j} + c^{(i_s)j} = 0, j \in I^{-(i)} \tag{63}$$

$$v^{(i)j} + c^{(i_s)j} - \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W^{(i_{s2})}_{:,j} = 0, j \in I^{+(i)} \tag{64}$$

$$\text{for } j \in I^{\pm(i)} : \tag{65}$$

$$v^{(i)j} + c^{(i_s)j} + \tau^{(i_s)j} - \lambda^{(i_s)j}s^{(i_s)j} + \pi^{(i_s)T}P^{(i_s)}_{:,j} = 0 \tag{66}$$

$$\lambda^{(i_s)j} - (\mu^{(i_s)j} + \tau^{(i_s)j}) = \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W^{(i_{s2})}_{:,j} - \pi^{(i_s)T}\hat{P}^{(i_s)}_{:,j} \tag{67}$$

As any valid assignments of v and $\gamma, \mu, \lambda, \pi, \tau \ge 0$ lead to a safe bound and we aim to cut down the number of Lagrangian variables to save the overhead, we declare and assign some of the variables following the convention in [1]:

$$\text{for } i \in [L_1] \cup [L_2] \text{ and } i_s \in S(i) \cap [R], j \in I^{\pm(i)} : \tag{68}$$

$$\hat{v}^{(i_s)T} = \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W^{(i_{s2})}_{:,j} - \pi^{(i_s)T}\hat{P}^{(i_s)}_{:,j} \tag{69}$$

$$\lambda^{(i_s)j} = [\hat{v}^{(i_s)T}]_+ \tag{70}$$

$$\mu^{(i_s)j} + \tau^{(i_s)j} = [\hat{v}^{(i_s)T}]_- \tag{71}$$

$$\tau^{(i_s)j} = \alpha^{(i_s)j} \cdot [\hat{v}^{(i_s)T}]_- \tag{72}$$

$$\text{s.t. } \alpha^{(i_s)j} \in [0, 1] \tag{73}$$

At line 70 and 71, $[x]_+ = \max(x, 0), [x]_- = -\min(x, 0)$, both resulting in a non-negative result. After replacement of line 69-71 at Equation 66 and 67, we have the following variable assignments from constraints 59-67:

$$v^{(L)T} = -H^T\gamma \tag{74}$$

$$\text{for } i \in [L_1] \text{ and } i_s \in S(i) \cap [L_2] \text{ and } i_s \notin [R] : \tag{75}$$

$$v^{(i)} = v^{(i_s)} \tag{76}$$

$$\text{for } i \in [L_1] \cup [L_2] \text{ and } i_s \in S(i) \cap [R] : \tag{77}$$

$$v^{(i)j} = -c^{(i_s)j}, j \in I^{-(i)} \tag{78}$$

$$v^{(i)j} = \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W^{(i_{s2})}_{:,j} - c^{(i_s)j}, j \in I^{+(i)} \tag{79}$$

$$\text{for } j \in I^{\pm(i)} : \tag{80}$$

$$\hat{v}^{(i_s)T} = \sum_{i_{s2}\in S(i_s)\cap[L_1]} v^{(i_{s2})T}W^{(i_{s2})}_{:,j} - \pi^{(i_s)T}\hat{P}^{(i_s)}_{:,j} \tag{81}$$

$$v^{(i)j} = [\hat{v}^{(i_s)T}]_+ s^{(i_s)j} - c^{(i_s)j} - \pi^{(i_s)T}P^{(i_s)}_{:,j} \tag{82}$$

$$-\alpha^{(i_s)j} \cdot [\hat{v}^{(i_s)T}]_-, \text{ s.t. } \alpha^{(i_s)j} \in [0, 1] \tag{83}$$

By substituting variables at formula 45 - 57 with the above-mentioned variable assignments and minimization at Equation 58, we reap the unconstrained problem stated at Equation 2-13. □

References

[1] H. Zhang, S. Wang, K. Xu, L. Li, B. Li, S. Jana, C. Hsieh, and J. Z. Kolter, “General cutting planes for bound-propagation-based neural network verification,” in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., 2022. [Online]. Available: http://papers.nips.cc/paper_files/paper/2022/hash/0b06c8673ebb453e5e468f7743d8f54e-Abstract-Conference.html