1 Suggestions

1. In this chapter, the complexity classes FPT, $W[1]$, and $W[2]$ are introduced by going through harder and harder problems (in terms of parametrized complexity) and trying to find lower bounds on hardness. I definitely found this presentation very helpful compared to other texts as it motivated the construction of these classes. However, I felt like the introduction of the k-step Nondeterministic Turing Machinery (k-step-NTM) may not need to be so early and could be skipped till later. Specifically, I think you can straight away define W[1] as the complexity class for which Clique and IS are complete. The fact that parametrized Clique and IS have no known polynomial algorithm already motivates the construction of such a parametrized complexity class that is harder than FPT and complete for Clique and IS. You have done something similar to what I am suggesting in the next section, where you have defined the complexity class W[2] as the class which is complete for Dominating Set solely based on the intuition that Dominating Set is harder than Clique and IS (but without using the W[2] complete Turing Machinery problem - "k-step-NTM with multiple tapes"). I think that it would be more clear if the Turing machine problems were listed later on or in a separate section as explained in the next point

2. In this chapter, the parametrized complexity classes are first introduced through their complete problems. Then, towards the end, you introduced an equivalent definition of these complexity classes through properties of their circuit representations. I think you can add another section after, where you introduce another equivalent definition of these complexity classes in terms of variants of the Turing machine halting problems, i.e.

   - W[1]: k-step-NTM
   - W[i]: k-step-multi-NTM (multiple (i) tapes)
   - W[P]: Bounded-NTM (explained in suggested problem 11)
   - W[SAT]: I assume there may exist some variant of the halting problem that is complete for this class, but I do not know

3. Is it true that all problems in FPT are kernelizable? This is not clear from the text because FPT is defined after kernelization, and it is defined without reference to kernelization. I only was able to infer this due to the comment at the end of theorem 8.2.1: "this will soon be called Fixed Parameter Tractable." I think you should either switch the order of 8.2 and 8.3 (because the definition of FPT immediately follows the discussion in 8.1), or you should make this another lemma/theorem

4. I think you should refer to all problems with "k-" as a prefix for clarity (i.e. "k-clique")

5. It would be useful to have a diagram/chart of the different problems in each class of the W heirarchy (similar to my list in the summary)

6. I think it would be useful to discuss/mention a problem which is outside of XP and in NP

7. In some other texts, k-step-NTM is refered to as a "halting problem" or an "acceptance problem," for example k-step-Turing Machine Acceptance. I, personally, would find this labelling more recognizeable and understandable than "Turing Machinery"

8. FPT is the class of problems that can be solved in $O(f(k)n^{O(1)})$ time while XP is the class of problems that can be solved in $O(f(k)n^{O(g(k))}$. I found myself curious what is the runtime big-O of the intermediate classes W[1] and W[2]. Is this well understood? If so, it would be useful to mention this.

2 Comments for Improvements on the Chapter

2.1 Regarding Theorem 8.1.1

When going through the proof, I understood it but I believe it can be slightly modified without becoming more complicated, to be more easily understood by the reader.
Specifically, I think that points 3 and 4. of the proof are not very clear. For example, point 3. might never be satisfied, since there might be a VC of size $< k$. I believe that point 3. should be:
"3. Keep doing this until either the tree is of height k or there are no edges left in the set $G - R$, where $R \subseteq G$ is the set of vertices removed by this path of the algorithm's tree so far."
Similarly, point 4. should be:
"If one of the leaves' graph $G - R$ contains no edges, then R is a vertex cover of size $\leq k$. If not, then there is not."

2.2 Chapter Bugs/Improvements

Since some improvements I suggest can be also considered bugs, I added this section, where I explain them.

1. On page 213, in the proof of Theorem 8.2.1, on step 3, it should be "If there is a vertex v of degree ***at least*** $L + 1 \ldots$". The algorithm does not work properly with the exact value.

2. On page 214, Theorem 8.2.1 should be denoted Theorem 8.2.4.

3. Page 215, Ch.8.5, question mark missing in first sentence of second paragraph.

3 Improving Figure 8.1

Using Tikz I drew in LaTeXthe sketched figure 8.1. Note that it is easy to modify the figure to fit it later optimally in the chapter in the way you prefer.

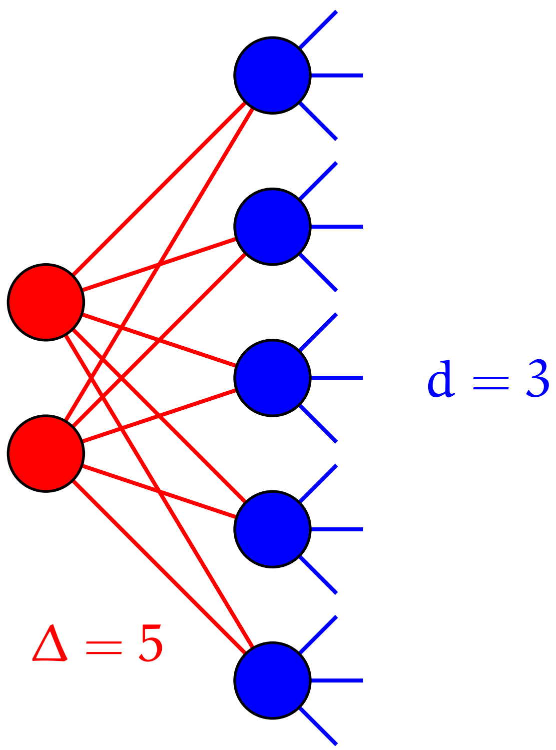

Figure 1: Reduction of Cliq to regular-Cliq