Spanning trees in graphs of high minimum degree with a universal vertex I: An asymptotic result

Bruce Reed1 | Maya Stein2

1Department of Mathematics and Statistics, University of Victoria, Victoria, British Columbia, Canada
2Department of Mathematical Engineering, University of Chile, Santiago, Chile

Correspondence
Maya Stein, Department of Mathematical Engineering, University of Chile, Beauchef 851, Santiago, Región Metropolitana, Chile. Email: mstein@dim.uchile.cl

Abstract
In this paper and a companion paper, we prove that, if $m$ is sufficiently large, every graph on $m + 1$ vertices that has a universal vertex and minimum degree at least $\left\lceil \frac{2m}{3} \right\rceil$ contains each tree $T$ with $m$ edges as a subgraph. Our result confirms, for large $m$, an important special case of a recent conjecture by Havet, Reed, Stein and Wood. The present paper already contains an approximate version of the result.

KEYWORDS
graph, maximum degree, minimum degree, spanning tree

1 | INTRODUCTION

A recurring topic in extremal graph theory is the use of degree conditions (such as minimum/average degree bounds) on a graph to prove that it contains certain subgraphs. One of the easiest classes of subgraphs for which this question is not yet properly understood is trees. This is the focus of the present paper.

Clearly, any graph of minimum degree exceeding $m − 1$ contains a copy of each tree with $m$ edges: Just embed the root of the tree anywhere in the host graph, and greedily continue, always embedding vertices whose parents have already been embedded. The bound on the minimum degree is sharp (see below).

Our paper is one of a large number which discusses possible strengthenings of the above observation by replacing the minimum degree condition with a different condition on the degrees of the host graph. One of these is the Loebl–Komlós–Sós conjecture from
1995 (see [7]), which replaces the minimum degree with the median degree. This conjecture has attracted a fair amount of attention over the last decades, and has been settled asymptotically [11–14].

More famously, Erdős and Sós conjectured in 1963 that every graph of average degree exceeding \( m - 1 \) contains each tree with \( m \) edges as a subgraph. This conjecture would be best possible, since no \((m - 1)\)-regular graph contains the star \( K_{1,m} \) as a subgraph. Alternatively, one can consider a graph that consists of several disjoint copies of the complete graph \( K_m \); this graph has no connected \((m + 1)\)-vertex subgraph at all. Note that for both of these examples it does not matter whether we considered the average degree (as in the Erdős–Sós conjecture) or the minimum degree (as in the observation above).

The Erdős–Sós conjecture poses an extremely interesting question. It is trivial for stars, and it holds for paths by an old theorem of Erdős and Gallai [8]. It also holds when some additional assumptions on the host graph are made, see, for instance, [5, 10, 20, 21], and there are versions for bounded degree trees [4, 19]. In the early 1990s, Ajtai, Komlós, Simonovits and Szemerédi announced a proof of the Erdős–Sós conjecture for sufficiently large \( m \).

It is well known that every graph of average degree \( >m \) has a subgraph of minimum degree \( >\frac{m}{2} \). So, if it were true that every graph of minimum degree exceeding \( \frac{m}{2} \) contained each tree on \( m \) edges, then the Erdős–Sós conjecture would immediately follow. Of course, the statement from the previous sentence is not true: It suffices to consider the examples given above. Still, for bounded degree spanning trees an approximate version of the statement does hold. Komlós, Sárközy and Szemerédi show in [15] that every large enough \((m + 1)\)-vertex graph of minimum degree at least \((1 + \delta)\frac{m}{2}\) contains each tree with \( m \) edges whose maximum degree is bounded by \( \frac{cn}{\log n} \), where the constant \( c \) depends on \( \delta \). Variations of the bounds and the size of the tree are given in [2, 6]. However, the result from [15] is essentially the best possible in the sense that (even if the minimum degree of the host graph is raised) it does not hold for trees of significantly larger maximum degree [15].

So, if we wish to find a condition that guarantees we can find all trees of a given size as subgraphs, only bounding the minimum degree is not enough. Nevertheless, there can be at most one vertex of degree at least \( \frac{m}{2} \) in any tree on \( m + 1 \) vertices, and so, we might not need many vertices of large degree in the host graph. Therefore, it seems natural to try to pose a condition on both the minimum and the maximum degree of the host graph.

The first conjecture of this type has been put forward recently by Havet, Reed, Stein and Wood [9]. They believe that a maximum degree of at least \( m \) and a minimum degree of at least \( \left\lfloor \frac{2m}{3} \right\rfloor \) is enough to embed all \( m \)-edge trees.

**Conjecture 1.1** (Havet, Reed, Stein and Wood [9]). Let \( m \in \mathbb{N} \). If a graph has a maximum degree at least \( m \) and a minimum degree at least \( \left\lfloor \frac{2m}{3} \right\rfloor \) then it contains every tree with \( m \) edges as a subgraph.
The conjecture holds if the minimum degree condition is replaced by \((1 - \gamma)m\), for a tiny but explicit\(^1\) constant \(\gamma\), and it also holds if the maximum degree condition is replaced by a large function\(^2\) in \(m\) [9].

Moreover, an approximate version of the conjecture holds for bounded degree trees and dense host graphs [2]. Such an approximate version even holds for a generalised form of Conjecture 1.1, where the bound on the minimum degree is allowed to be any value between \(\frac{m}{2}\) and \(\frac{2m}{3}\), with the maximum degree obeying a corresponding bound between \(2m\) and \(m\) (see [3] for details).

As further evidence for Conjecture 1.1 we shall prove, in this paper and its companion paper [18], that the conjecture holds for sufficiently large \(m\), under the additional assumption that the host graph has \(m + 1\) vertices, that is, we are looking for a spanning tree. That is, building on the results from the present paper, we will show the following theorem in [18].

**Theorem 1.2** (Reed and Stein [18]). There is an \(m_0 \in \mathbb{N}\) such that for every \(m \geq m_0\) every graph on \(m + 1\) vertices which has a minimum degree at least \(\left\lfloor \frac{2m}{3} \right\rfloor\) and a universal vertex contains every tree \(T\) with \(m\) edges as a subgraph.

Observe that Theorem 1.2 is easy if \(T\) has a vertex \(t\) that is adjacent to a set \(L\) of at least \(\delta m\) leaves. Indeed, we root \(T\) at \(t\), embed \(t\) in the universal vertex \(v^*\) of \(G\), greedily embed \(T - L\), and then embed \(L\) in neighbours of \(v^*\). This is possible since \(v^*\) is universal. It turns out that this approach can be extended if, for a small positive number \(\delta\), the tree \(T\) contains a vertex adjacent to at least \(\delta n\) leaves. Although the greedy argument no longer works, we will be able to prove a result, namely, Lemma 1.3, which achieves the embedding of any tree \(T\) as above. This lemma will be crucial for the proof of Theorem 1.2 in our companion paper [18].

**Lemma 1.3.** For every \(\delta > 0\), there is an \(m_\delta\) such that for any \(m \geq m_\delta\) the following holds for every graph \(G\) on \(m + 1\) vertices which has a minimum degree at least \(\left\lfloor \frac{2m}{3} \right\rfloor\) and a universal vertex. If \(T\) is a tree with \(m\) edges, and some vertex of \(T\) is adjacent to at least \(\delta m\) leaves, then \(G\) contains \(T\).

Also, the results from the present paper already imply an asymptotic version of Theorem 1.2.

**Theorem 1.4.** For every \(\delta > 0\), there is an \(m_\delta\) such that for every \(m \geq m_\delta\), every graph \(G\) on \(m + 1\) vertices having a minimum degree at least \(\left\lfloor \frac{2m}{3} \right\rfloor\) and a universal vertex contains every tree \(T\) having at most \((1 - \delta)m\) edges.

Both Theorem 1.4 and Lemma 1.3 follow quickly from Lemma 2.1, which is stated in Section 2. The proof of Lemma 2.1 occupies almost all the remainder of this paper, namely, Sections 4–6, and a sketch of this proof is given in Section 3.

---

\(^1\)Namely, \(\gamma = 200^{-30}\).

\(^2\)Namely, \(f(m) = (m + 1)2^{m+6} + 1\).
In the companion paper \[18\], we prove the full Theorem 1.2, building on Lemma 1.3 and on another auxiliary result, namely, Lemma 7.3, which is similar to Lemma 1.3. Lemma 7.3 is stated and proved in the last section of the present paper, Section 7.

Let us end the introduction with a very short sketch of our methods of proof. A more detailed overview can be found in Section 3.

Given a tree $T$ we wish to embed in the host graph $G$, we first cut $T$ into a constant number of connecting vertices, and a large number of very small subtrees. Applying regularity to $G$, we can ensure that all small trees that are not just leaves can be embedded into matching structures we find in the reduced graph. This is more complicated than in earlier work on tree embeddings using the regularity approach, as our assumptions are too weak to force one matching structure we can work with throughout the whole embedding. Instead, we have to employ ad hoc matchings, plus some auxiliary structures, one for each of the connecting vertices. Finally, we have to deal with the leaves adjacent to connecting vertices. These are more difficult to embed than the other small trees, because an embedded vertex might only see two-thirds of the graph, and there is no way to reach the remaining third of the graph in only one step. For this reason, we have to come up with a delicate strategy on where we place the connecting vertices, to ensure that at the very end of the embedding process we will be in a position to embed all these leaves at once with a Hall-type argument.

2 | THE PROOF OF LEMMA 1.3 AND THEOREM 1.4

The lemma behind our two results from the introduction (Lemma 1.3 and Theorem 1.4) is the following.

**Lemma 2.1.** For every $\delta > 0$, there is an $m_\delta$ such that for any $m \geq m_\delta$ and $\alpha$ with $\delta \leq \alpha \leq 1$ the following holds.

Let $G$ be an $(m + 1)$-vertex graph of minimum degree at least $\lfloor \frac{2m}{3} \rfloor$, and let $w \in V(G)$. Let $T$ be a tree with at most $(1 - \alpha)m$ edges, let $t \in V(T)$, and assume that no vertex of $T$ is adjacent to more than $\alpha m$ leaves. Then one can embed $T$ in $G$, mapping $t$ to $w$.

Let us see how Lemma 2.1 implies the results from the introduction. Here is the proof of the lemma that we will need in the companion paper [18].

**Proof of Lemma 1.3.** Let $m_\delta$ be given by Lemma 2.1 for input $\delta$. Given $G$ and $T$ as in Lemma 1.3, we let $t$ be a vertex of $T$ having the maximum number of leaf neighbours. Let $L$ be the set of its leaf neighbours and set $\alpha := \frac{|L|}{m}$. By assumption, $\delta \leq \alpha \leq 1$, so we may apply Lemma 2.1 to obtain an embedding of $T - L$ in $G$ with $t$ embedded in the universal vertex of $G$. We can then embed the vertices of $L$ into the remaining vertices of $G$. $\square$

Here is the proof of the approximate result.
**Proof of Theorem 1.4.** Let $m_δ$ be the maximum of the numbers $m_δ$ given by Lemma 2.1 and by Lemma 1.3 for input $δ$. Given $G$ and $T$ as in the theorem, consider a vertex of $T$ with the maximum number of leaf neighbours, say these are $βm$ leaf neighbours. If $β ≤ δ$, we are done by Lemma 2.1. If $β > δ$, we are done by Lemma 1.3. □

3 | A SKETCH OF THE PROOF OF LEMMA 2.1

The purpose of this section is to give some more detailed insight into the proof of Lemma 2.1, going a little more below the surface than in the Introduction. We remark that for the understanding of the rest of the paper, it is not necessary to read this section (but we hope it will be helpful).

The number $m_δ$ is chosen in dependence of the output of the regularity lemma for some constant depending on $δ$. Now, given the approximation constant $α$, the tree $T$ and the host graph $G$, we prepare each of $T$ and $G$ separately for the embedding.

Similar as in earlier tree embedding proofs [14, 17], we cut $T$ into a set $W$ of seeds (connecting vertices), such that $W$ has constant size, and a large set $T$ of very small subtrees. The trees in $T$ are only connected through $W$, and they each have size $< βm$, where $β$ is a small constant (smaller than all other constants in this paper). Differently from earlier approaches to tree cutting, we now categorise the small trees contained in $T$. They fall into three categories: trees consisting of a leaf of $T$, trees that are smaller than a (huge) constant, and trees that are larger than this constant. We name the categories $LF_1$, $F_2$. The last category is further subdivided into two sets, $F'_2$ and $F''_2$, according to whether the small tree is adjacent to one or more of the seeds. Each seed from $W$ may have trees of any (or all) of these categories hanging from it (and there may also be seeds hanging from it). The details of this cut-up of $T$ are explained in Section 5.1.

Next, in Section 5.2, we order and group the seeds obtained from this decomposition. Our strategy of ordering the seeds takes into account their position in a natural embedding order, but also the number of leaves hanging from them. We will come back to this point at a later stage during this outline, and will then explain the why and how of the ordering.

Independently, in Section 6.1.1, we regularise the host graph $G$, with parameter $ε$, such that $β ≪ ε ≪ α$. (For an introduction to regularity, see Section 4.3.) Furthermore, we partition each of its clusters $C$ arbitrarily into subsets $C_W$, $C_L$, $C_{F_1}$, $C_{F_2}$, $C_{V'}$ of appropriate sizes into which we aim to embed the different parts of the tree, namely, $W$, $L$, $F_1$ and $F_2$, while the last subset, $C_{V'}$, is reserved for neighbours of seeds in trees of $F'_2$. The set of these neighbours will be denoted by $V'$.

We fix a matching $M_{F_2}$ in the reduced graph $R_G$. This matching will be used when we embed the trees from $F_2$. More precisely, we will embed each tree $T ∈ F_2$ into $C_{F_2} ∪ D_{F_2}$ for a suitable (i.e., sufficiently unoccupied) edge $CD ∈ M_{F_2}$, except for the root $r_T$ of $T$. The root $r_T$ will go to one of the subsets $C'_τ$, for a suitable cluster $C'$ that connects $CD$ with the cluster containing the seed adjacent to $T$. In case $T ∈ F'_2$, which means that $T$ contains a second vertex $v$ from $V'$, we embed $v$ into one of the subsets $C''_τ$, for a suitable cluster $C''$. Throughout the embedding process, we will keep each of the edges of $M_{F_2}$ as balanced as possible. That is, the sets of used vertices in the corresponding slices $C_{F_2}$ or $D_{F_2}$ on either side of such an edge never differ by more than $βm$.  


Since we do not have enough space in the slices $C_V$ for all roots of trees in $F_1$ (because we have no control over the number of trees in $F_1$), we need to proceed differently with the small trees from $F_1$. For embedding these trees, we use a family of matchings $M_s$, one for each embedded seed $s$. Since these matchings $M_s$ are possibly different for each $s$, we now will have to keep the set of all slices $C_{F_1}$ balanced. This is not easy but possible since the trees from $F_1$ have constant size, and we choose $M_s$ so that it intersects the neighbourhood of the image of $s$ in a nice way. In addition to $M_s$, we also employ two auxiliary matchings which we combine with $M_s$ to obtain a partition of almost all of $V(R_G)$ with short paths. We call these structures good path partitions and, together with the matchings $M_s$, they will be defined and proved to exist in Section 4.2.

The actual embedding of the tree will be performed as follows. In Sections 6.2 and 6.3, we go through the seeds in a connected way, and embed each seed $s$ together with all the trees from $F_1 \cup F_2$ hanging at $s$ in the corresponding slices in the way we discussed above. We leave out any leaves from $L$, as we will deal with them in the final phase of the embedding. Our precautions from above ensure that we can embed all of $T - L$ without a problem, never running out of space. However, if we do not take care where exactly we embed the seeds, we may run into problems in the final phase when we need to embed the leaves. For instance, we need to avoid embedding all parents of $L$ into vertices having the same neighbourhood in $G$, as then, the leaves might not fit.

For this reason, we take some extra care when choosing the target clusters and the images for the seeds (this happens in Section 5.2). As already shortly mentioned above, we order the seeds into a system of groups according to the number of leaves hanging from them, and also according to the order the seeds appear in our planned embedding order. Moreover, each seed $s$ will be assigned a relevant set $X_s$ of seeds that come before it. In the actual embedding, in Section 6.2, we choose the image $\varphi(s)$ of a given seed $s$ in a way that $\varphi(s)$ has many neighbours outside the union of the neighbourhoods of $\varphi(X_s)$. (We remark that it is crucial here that no vertex of $T$ is adjacent to more than $\alpha m$ leaves.) This precaution will ensure that for each subset of seeds, their images have enough neighbours in $Z := \bigcup C_L$. We will then be able to embed all the leaves in $L$ at once by using Hall’s theorem. The whole procedure will be explained in detail in Section 6.4.

The last section of this paper, Section 7, is devoted to the proof of Lemma 7.3, which will need Lemma 7.3 in our companion paper [18]. The lemma deals with a similar situation as the one treated in Lemma 2.1, the main difference being that now, a small part of the tree is already embedded (and thus possibly blocking valuable neighbourhoods), but, on the positive side, throughout [18], we will be able to assume that no seed is adjacent to many leaves, and so we can assume this as well in Lemma 7.3.

4 | PRELIMINARIES

4.1 | An edge-double-counting lemma

We will need the following easy lemma.

**Lemma 4.1.** Let $G$ be a graph on $n$ vertices, let $0 < \psi < \frac{1}{3}$, and let $S \subseteq V(G)$. If each vertex in $S$ has degree at least $\left(\frac{2}{3} - \psi\right)n$, then there are at least $\left(\frac{1}{3} + \frac{\sqrt{\psi}}{10}\right)n$ vertices in $G$ that each has at least $\left(\frac{1}{2} - \sqrt{\psi}\right)|S|$ neighbours in $S$. 
Proof. Let \( A \subseteq V(G) \) denote the set of all vertices having at least \( \left( \frac{1}{2} - \sqrt{\psi} \right) |S| \) neighbours in \( S \). Writing \( e(S, V(G)) \) for the number of all edges touching \( S \), where edges inside \( S \) are counted twice, we calculate that

\[
\left( \frac{2}{3} - \psi \right) n \cdot |S| \leq e(S, V(G)) \\
\leq |V(G) \setminus A| \cdot \left( \frac{1}{2} - \sqrt{\psi} \right) |S| + |A| \cdot |S| \\
\leq n \cdot \left( \frac{1}{2} - \sqrt{\psi} \right) |S| + |A| \cdot \left( \frac{1}{2} + \sqrt{\psi} \right) |S|,
\]

and conclude that

\[
|A| \geq \frac{1}{6} + \frac{\sqrt{\psi}}{2} \cdot n \geq \left( \frac{1}{3} + \frac{\sqrt{\psi}}{10} \right) \cdot n,
\]

as desired. \( \square \)

4.2 Matchings and good path partitions

The purpose of this subsection is to find some matchings in a graph \( H \) (which will later be the reduced graph \( R_G \) of our host graph \( G \)), and combinations of some of these matchings to covers of \( H \) with short paths. These structures will be used for the embedding of \( T \) in the proof of Lemma 2.1, more specifically in Section 6.3. The important result of this section is Lemma 4.3.

We need a quick definition before we start. For any graph \( H \), and any \( \subseteq N V_H \), a \( N \)-good matching is one whose edges each have at most one vertex outside \( N \).

Lemma 4.2. Let \( 0 < \xi < \frac{1}{20} \) and let \( H \) be a \( p \)-vertex graph of minimum degree at least \( \left( \frac{2}{3} - \xi \right)p \). Let \( N \subseteq V(H) \) be such that \(|N| = \left\lfloor \left( \frac{2}{3} - 2\xi \right)p \right\rfloor \). Then \( H - N \) contains a set \( Y \) of size at most \( 3\xi p + 1 \) such that \( H - Y \) has an \( N \)-good perfect matching.

Proof. First of all, note that we can greedily match all but a set \( X \) of at most \( 3\xi p \) vertices from \( V(H) \setminus N \) to \( N \), simply because of the condition on the minimum degree. If necessary, add one more vertex to \( X \), to have that \( H - X \) is even. So \(|X| \leq 3\xi p + 1 \). Now, take any maximal \( N \)-good matching \( M \) in the graph \( H \) that covers all vertices of \( V(H) \setminus (N \cup X) \). We would like to see that \( M \) covers all of \( H - X \), so for contradiction assume that \( N \setminus V(M) \) contains at least two vertices.

By the maximality of \( M \), we know that \( N \setminus V(M) \) is an independent set, and no two vertices \( D, D' \in N \setminus V(M) \) can be adjacent to different endpoints of an edge in \( M \). So, for each edge \( EF \in M \), we know that either one of the endvertices, say \( E \), sees no vertex in \( N \setminus V(M) \), or \( E \) and \( F \) each see only one vertex in \( N \setminus V(M) \) (and that is the same vertex). Therefore, at least one of the vertices in \( N \setminus V(M) \) sees at most half of the vertices in
V(M), and thus less than half of the vertices in H − X, a contradiction to our condition on the minimum degree.

Before we state the second lemma of this section, we need another definition. An $N$-out-good path partition of a graph $H$, with $N \subseteq V(H)$, is a set $\mathcal{P}$ of disjoint paths, together covering all the vertices of $H$, such that for each $P \in \mathcal{P}$ one of the following holds:

- $P = AB$, with $A, B \in N$;
- $P = ABCD$, with $B, C \in N$; or
- $P = ABCDE$, with $B, C, D, E \in N$.

(Note that if $P$ has four vertices, then there is no restriction on the whereabouts of $A$ and $D$, and similar for six-vertex paths $P$.)

An $N$-in-good path partition of a graph $H$, with $N \subseteq V(H)$, is a set $\mathcal{P}$ of disjoint paths, together covering all the vertices of $H$, such that for each $P \in \mathcal{P}$ one of the following holds:

- $P = AB$, with $A, B \in N$;
- $P = ABCD$, with $A, D \in N$; or
- $P = ABCDE$, with $A, C, D, F \in N$.

Now we are ready to state the main result of this section. Note that the first item is a direct consequence of the previous lemma, and all structures exist independently of each other.

**Lemma 4.3.** Let $0 < \xi < \frac{1}{20}$, and let $H$ be a $p$-vertex graph of minimum degree at least $(\frac{2}{3} - \xi)p$. Let $N \subseteq V(H)$ be any set with $|N| = \lceil (\frac{2}{3} - 2\xi)p \rceil$. Then $H$ contains a set $X$ of at most $\xi p^2 + 3$ vertices, such that

- $H \setminus X$ has an $N(X)$-good perfect matching;
- $H \setminus X$ has an $N(X)$-in-good path partition and
- $H \setminus X$ has an $N(X)$-out-good path partition.

**Proof:** Lemma 4.2 provides us with a set $Y$ and an $N$-good perfect matching $M$ of $H \setminus Y$. Note that $M$ would be as desired for the first item, if $X$ was chosen as $Y$.

Now, set $Q := V(M) \setminus N$, and let $R$ be the set of all vertices from $H$ that are matched by $M$ to a vertex from $Q$. Since $M$ is $N$-good, we know that $R \subseteq N$. Note that

$$|N \setminus R| \geq |N| - |Q| \geq |N| - (p - |N|) = 2|N| - p \geq \left(\frac{1}{3} - 4\xi\right)p.$$  \hspace{1cm} (1)

We take a maximal matching $\tilde{M}Q$ inside $H[Q]$. Let $\tilde{Q}$ be the set of vertices in $Q$ not covered by $\tilde{M}Q$. We augment $\tilde{M}Q$ to a matching $MQ$ by matching as many vertices of $\tilde{Q}$ as possible to vertices of $N \setminus R$. Let $Z$ denote the set of remaining vertices of $\tilde{Q}$. Note that each vertex $v \in Z$ sees at most one of the endvertices of any edge of $MQ \setminus \tilde{M}Q$, as $\tilde{Q}$ is
independent, and moreover, \( v \) does not see \( (N \setminus R) \setminus V(M^Q) \). Therefore, for each \( v \in Z \), we have

\[
\left( \frac{2}{3} - \xi \right) p \leq |N(v)| \leq |Q \setminus Z| + |V(M^Q)| + |Y| + |R| = |Q \setminus Z| + |Y| + |R| = p - |N \setminus R| - |Z|,
\]

and hence, by (1), we have

\[
|Z| \leq \left( \frac{1}{3} + \xi \right) p - |N \setminus R| \leq 5 \xi p.
\]

Let \( Z' \subseteq R \) be the set of all vertices matched to vertices of \( Z \) by \( M \).

We define a second auxiliary matching \( M^R \) in a very similar way. Namely, \( M^R \) consists of edges with both ends in \( R \setminus Z' \), and a matching of almost all the remaining vertices of \( R \setminus Z' \) to vertices of \( N \setminus R \). We now take some more care: For each pair of vertices \( v, v' \in R \setminus Z' \) whose partners \( w, w' \) in \( M \) form an edge in \( M^Q \), we will match either both of \( v, v' \) or none (in each of the two matching steps). Letting \( Z'' \) denote the set of all unmatched vertices of \( R \setminus Z' \), we can calculate similarly as above that for each vertex \( v \) from \( Z'' \), either \( v \) or the paired vertex \( v' \) has degree at most \( 5 \xi p + 1 \). This leads to the bound \( |Z''| \leq 5 \xi p + 1 \).

Let \( Z''' \subseteq Q \) denote the set of all partners of vertices of \( Z'' \) in \( M \). Set \( X := Y \cup Z \cup Z' \cup Z'' \cup Z''' \). Note that

\[
|X| \leq 3 \xi p + 1 + 2 \cdot 5 \xi p + 2 \cdot (5 \xi p + 1) = 23 \xi p + 3.
\]

Now, discard any edge that touches \( X \) from \( M, M^Q \) and \( M^R \), thus obtaining matchings \( \overline{M}, \overline{M}^Q \) and \( \overline{M}^R \). Then the union \( U \) of the edges in \( \overline{M} \cup \overline{M}^Q \) gives an \((N \setminus X)\)-in-good path partition of \( H - X \). Indeed, consider a component \( P \) of \( U \).

First assume \( P \) contains an edge \( BC \) from \( \overline{M}^Q \). If \( B, C \in Q \), then (by the choice of \( M^Q \)), we know that \( \overline{M} \) contains edges \( AB \) and \( CD \), with \( A, D \in R \). The only other case (modulo renaming \( B \) and \( C \)) is that \( B \in Q \), and \( C \in N \setminus R \). Then again (by definition of \( M^Q \)), we know that \( \overline{M} \) contains edges \( AB \) and \( CD \), with \( A \in R \) and \( D \in N \setminus R \). If \( D \) is not matched in \( \overline{M}^Q \), then \( ABCD \) is as desired for the path partition. If \( D \) is matched in \( \overline{M}^Q \), say to a vertex \( E \in Q \setminus X \), then \( ABCDE \) is as desired for the path partition, where \( F \) is the partner of \( E \) in \( \overline{M} \).

Now assume \( P = AB \) is an edge from \( \overline{M} \). If \( A, B \in N \), then \( P \) is as desired, so assume \( B \in Q \). Then \( B \) is matched to some vertex \( C \in Q \) in \( M^Q \), but \( BC \notin \overline{M}^Q \). Thus \( C \in Z''' \). However, because of the way we chose \( M^R \) and \( Z''' \), we know that \( A \) is matched in \( M^R \), which means that also the partner of \( C \) in \( M \) is matched in \( M^R \), and thus \( C \notin Z''' \), a contradiction.

Similarly, the union of the edges in \( \overline{M} \cup \overline{M}^R \) gives the desired \( N \)-out-good path partition. Finally, \( \overline{M} \) is an \((N \setminus X)\)-good matching of \( H - X \). \( \square \)
4.3 | Regularity

We need to quickly discuss Szemerédi’s regularity lemma and a couple of other preliminaries regarding regularity. Readers familiar with this topic are invited to skip this subsection.

The density of a pair \((A, B)\) of disjoint subsets \(A, B \subseteq V(G)\) is

\[
|d(A, B) - d(A', B')| < \varepsilon
\]

for all \(A' \subseteq A, B' \subseteq B\) with \(|A'| \geq \varepsilon|A|, |B'| \geq \varepsilon|B|\). In many ways, regular pairs behave like random bipartite graphs with the same edge density.

If \((A, B)\) is an \(\varepsilon\)-regular pair, then we call a subset \(A' \subseteq A\) \(\varepsilon\)-significant (or simply significant, if \(\varepsilon\) is clear from the context) if

\[
|A' - \mu(A)| < \varepsilon n
\]

for some \(\mu(A)\). We call a vertex from \(A\) \(\varepsilon\)-typical (or simply typical, if \(\varepsilon\) is clear from the context) with respect to a set \(B' \subseteq B\) if it has degree at least \((1 - \varepsilon)d(A, B)|B'|\) to \(B'\).

The following well-known and easy-to-prove facts (see, e.g., [16]) state that in a regular pair almost every vertex is typical to any given significant set, and also that regularity is inherited by subpairs. More precisely, if \((A, B)\) is an \(\varepsilon\)-regular pair with density \(d\), then

(R1) for any \(\varepsilon\)-significant \(B' \subseteq B\), all but at most \(|\varepsilon A|\) vertices from \(A\) are \(\varepsilon\)-typical to \(B'\) and

(R2) for each \(\delta \geq 0\), and for any subsets \(A' \subseteq A, B' \subseteq B\), with \(|A'| \geq \delta|A|\) and \(|B'| \geq \delta|B|\), the pair \((A', B')\) is \(\frac{2\varepsilon}{\delta}\)-regular with density between \(d - \varepsilon\) and \(d + \varepsilon\).

Szemerédi’s regularity lemma states that every large enough graph has a partition of its vertex set into one “trash” set of bounded size, and a bounded number of sets of equal sizes, such that almost all pairs of these sets are \(\varepsilon\)-regular.

Lemma 4.4 (Szemerédi’s regularity lemma). For each \(\varepsilon > 0\) and \(M_0 \in \mathbb{N}\) there are \(M_0, n_0 \in \mathbb{N}\) such that for all \(n \geq n_0\), every \(n\)-vertex graph \(G\) has a partition \(V_0 \cup V_1 \cup \cdots \cup V_p\) of \(V(G)\) into \(p + 1\) partition classes (or clusters) such that

(a) \(M_0 \leq p \leq M_0\);  
(b) \(|V_1| = |V_2| = \cdots = |V_p|\) and \(V_0 < \varepsilon n\);  
(c) apart from at most \(\varepsilon\left(\frac{p}{2}\right)\) exceptional pairs, the pairs \((V_i, V_j)\) are \(\varepsilon\)-regular, for \(i, j > 0\) with \(i \neq j\).

As usual, we define the reduced graph \(R_G\) corresponding to this decomposition of \(G\) as follows. The vertices of \(R_G\) are all clusters \(V_i\) \((i = 1, \ldots, p)\), and \(R_G\) has an edge between \(V_i\) and \(V_j\) if the pair \((V_i, V_j)\) is \(\varepsilon\)-regular, and has a density at least \(10\sqrt{\varepsilon}\). By standard calculations (see, e.g., [16]), and assuming we take \(M_0 \geq \lfloor \frac{1}{\varepsilon} \rfloor\), it follows that

\[
\delta_{w}(R_G) \geq (1 - 12\sqrt{\varepsilon}) \cdot \frac{p}{|V(G)|} \cdot \delta(G),
\]
where $\delta_w(R_G)$ is the weighted minimum degree. (That is, the densities of the pairs of clusters provide weights on the edges of $R_G$, and the weighted degree of a vertex is the sum of the corresponding edge-weights. The weighted minimum degree is the minimum of these weighted degrees. Observe that $\delta_w(R_G) \leq \delta(R_G)$ since weights do not exceed 1.)

Almost all vertices of any cluster $C \in V(R_G)$ are typical to almost all significant sets, in the following sense. If $\mathcal{Y}$ is a set of significant subsets of clusters in $V(R_G)$, then

$$\text{all but at most } \sqrt{\varepsilon |C|} \text{ vertices } v \in C \text{ are typical with respect to all but at most } \sqrt{\varepsilon |\mathcal{Y}|} \text{ sets in } \mathcal{Y}. \quad (3)$$

To see this well-known observation, assume that the set $\subseteq \mathcal{C}$ of vertices not satisfying (3) is larger than $\sqrt{\varepsilon |C|}$. Then

$$\sum_{Y \in \mathcal{Y}} |\{v \in C : v \text{ is not typical to } Y\}| \geq \sum_{v \in C'} |\{Y \in \mathcal{Y} : v \text{ is not typical to } Y\}| \geq |C'| \sqrt{\varepsilon |\mathcal{Y}|} > \varepsilon |C| \cdot |\mathcal{Y}|.$$  

Thus there is a $Y \in \mathcal{Y}$ such that more than $\varepsilon |C|$ vertices in $C$ are not typical to $Y$, a contradiction.

Regularity will help us when embedding small trees into a pair of adjacent clusters of $R_G$.

**Lemma 4.5.** Let $CD$ be an edge of $R_G$, and let $U \subseteq G$ with $C \setminus U, |D \setminus U| \geq \sqrt{\varepsilon |C|}$. Let $T$ be a tree of size $\leq \varepsilon |C|$ with root $r_T$.

Then $T$ can be embedded into $G$, with $T - r_T$ going to $(C \cup D) \setminus U$, and with $r_T$ going to any prescribed set of $\geq 2\varepsilon |C|$ vertices of $C$, or to any prescribed set of $\geq 2\varepsilon |C|$ vertices of $C'$, where $C'$ is any other cluster of $R_G$ that is adjacent to $D$.

**Proof.** We construct the embedding $\tilde{T}$ levelwise, starting with the root, which is embedded into a typical vertex of $(C \cup D) \setminus U$. At each step $i$ we ensure that all vertices of level $i$ are embedded into vertices of $C \setminus U$ (or $D \setminus U$) that are typical with respect to the unoccupied vertices of $D \setminus U$ (or $C \setminus U$). This is possible, because at each step $i$, and for each vertex $v$ of level $i$, by (R1), the degree of a typical vertex into the unoccupied vertices on the other side is at least $4\varepsilon |C|$, and there are at most $\varepsilon |C|$ nontypical vertices and at most $|T| \leq \varepsilon |C|$ already occupied vertices. \qed

## 5 \hfill PREPARING THE TREE

### 5.1 \hfill Cutting a tree

In this section, we will show how any tree $T$ can be cut up into small subtrees and few connecting vertices. The idea is that later, we can use regular pairs to embed many tiny trees.

We will make use of a procedure which in a very similar shape has already appeared in [14, 17], although there, no distinctions between the trees from $L, F_1, F_2$, were made. The resulting cut-up is given in the following statement.
Lemma 5.1. For any \( m \in \mathbb{N} \), for any tree \( T \) on \( m + 1 \) vertices, for any \( r \in V(T) \), and for any \( \beta > 0 \), there is a set \( W \subseteq V(T) \), and a partition \( T = L \cup F_1 \cup F_2 \) of the family \( T \) of components of \( T - W \), distinguishing a subset \( F'_2 \subseteq F_2 \), such that

(a) \( r \in W \);
(b) \( |W| \leq \frac{2}{\beta} \);
(c) if \( \beta m > 1 \) then each \( w \in W \) has a child in \( T \);
(d) \( |V(T)| = 1 \) for every tree \( T \in L \);
(e) \( 1 < |V(T)| \leq \frac{1}{\beta} \) for every tree \( T \in F_1 \);
(f) \( \frac{1}{\beta} < |V(T)| \leq \beta m \) for every tree \( T \in F_2 \);
(g) each \( T \in L \cup F_1 \cup (F_2 \setminus F'_2) \) has exactly one neighbour in \( W \);
(h) each \( T \in F'_2 \) has exactly two neighbours in \( W \) and
(i) \( |V| < 2\beta m \), where \( V \) is the set of all neighbours of vertices of \( W \) in \( \bigcup F'_2 \).

The vertices in \( W \) will be called the seeds of \( T \).

Proof. In a sequence of at most \( \frac{1}{\beta} \) steps \( i \), we define vertices \( w_i \) and trees \( T_i \) as follows. Set \( T_0 := T \). Now, for each \( i > 0 \), let \( w_i \in V(T_{i-1}) \) be a vertex at maximal distance from \( r \) (the root of \( T \)) such that the components of \( T_i - w_i \) that do not contain \( r \) each have size at most \( \beta m \). Delete \( w_i \) and all of these components from \( T_{i-1} \) to obtain \( T_i \). We stop when we reach \( r \), which will be the last vertex \( w_i \) to be defined.

Let \( W_0 \) be the union of all \( w_i \), and let \( T_0 \) be the set of all components of \( T - W_0 \). These two sets already fulfil items (a), (b) and (c). (To see (b), note that at each step \( i \), we cut off \( \beta m \) vertices. Hence we actually have that \( W_i \leq \frac{1}{\beta} \).

To obtain sets \( W, T \) that also fulfil items (g) and (h), we add some vertices to \( W_0 \) as follows. For each \( T \in \bigcup T_0 \) that has \( \ell > 2 \) neighbours \( v_1, v_2, ..., v_\ell \) in \( W_0 \), we add to \( W_0 \) a set of at most \( \ell - 1 \) vertices \( w'_j \) from \( V(T) \) that separate all \( v_i \)’s from each other. Note that these are at most \( \frac{1}{\beta} \) vertices in total (counting over all affected \( T \)), since each of the newly added vertices \( w'_j \) can be associated with one of the “old” vertices \( v_i \) from \( W_0 \) such that \( v'_j \) lies between \( v_i \) and \( r \). So, let \( T_1 \) be the family of the trees in \( T - W_1 \), the new sets \( W_1, T_1 \) still fulfil (a)–(c) (actually, we have that \( |W'|| \leq \frac{2}{\beta} \)). They furthermore have the property that each of the trees in \( \bigcup T_1 \) has at most two neighbours in \( W_1 \).

We modify our sets once more to ensure that only the large trees can have two seed neighbours. We proceed as follows. For each \( T \in \bigcup T_1 \) that has at most \( \frac{1}{\beta} \) vertices and is adjacent to two seeds \( w_1, w_2 \in W_1 \), we add to \( W_1 \) all vertices in \( V(T) \). In total, these are at most \( \frac{1}{\beta} \cdot |W_i| \leq \frac{2}{\beta^2} \) vertices. Call the new set of seeds \( W_1 \).

Defining \( T \) as the family of the trees in \( T - W \), and adequately dividing \( T \) into three sets \( L, F_1, F_2 \), and letting \( F'_2 \) be the appropriate subset of \( F_2 \), we obtain sets that fulfil all properties of the claim (where (i) follows directly from (f), (h) and the observation that \( |W| < 2|F'_2| < \frac{|T - r_1|}{\beta} = 2\beta m \) ).
5.2 | Ordering the seeds of a tree

To be able to choose well the clusters of \( V(R_G) \) into which we will embed the seeds other than \( r \) later on, we will define a convenient ordering of the seeds of a tree \( T \) with a cut-up as in Section 5.1. Together with this ordering we will define a set of relevant seeds \( X_s \) for each seed \( s \) of the tree, and ensure that the seeds in \( X_s \) come before \( s \) in the ordering.

The purpose of this ordering and the definition of the sets \( X_s \) is that later, when we embed the trees from \( L, F_1 \) and \( F_2 \) in \( G \), it will turn out that the smaller the trees, the harder they are to embed, with the most difficult ones being the trees from \( L \), that is, the leaves of \( T \) adjacent to seeds. An embedded seed has only degree \( \frac{2}{3}m \) in \( G \), of which a large part might already be used, so we have to plan ahead to avoid getting stuck when embedding the leaves. For this reason we have to choose very well into which clusters the seeds go, and the sets \( X_s \) will help us with this.

The reader might wish to skip the remainder of this unfortunately rather technical section at a first reading, because everything we do here is only necessary for the embedding of \( L \). Even the embedding of \( L \) can be followed with only a vague understanding of the definitions of the present section if the reader takes the “Degrees of the embedded seeds into \( Z \)” as stated in Section 6.2.4 for granted.

5.2.1 | Grouping and ordering

Let us start with the ordering. Assume we are given a tree \( T \) which has been treated by Lemma 5.1 for some \( \beta > 0 \), let \( W \) denote the set of seeds we obtained. Throughout the rest of this section we will assume that

\[
|W| = 47 \cdot 2^{j^*}, \quad \text{where } j^* = \left\lfloor \log_4 \frac{2}{\beta^2} \right\rfloor. \tag{4}
\]

(This can be assumed by adding some new vertices to the tree \( T \), and declaring all of them seeds. We will explicitly discuss why this can be done in Section 6.1.2.)

We will order the seeds in two different ways, before we get to the third and final order. The first order is determined by the number of leaves hanging from each seed, the second order is determined by the position of the seeds in the tree \( T \), and the third order is a mixture of both. We explain the orderings in detail in the following.

We start by ordering the seeds in a way that the number of leaf children of the seeds is decreasing, and we call this the size order \( \sigma \) on the seeds. Now, we will define ordered sets of seeds, which we will call groups of seeds. First we will define the large groups: loosely speaking, the largest of these groups consists of all seeds, then we define two groups consisting of the first and the second half of the seeds respectively, then each of these groups is divided into two new groups, and so on, until the size is down to 47. More precisely, for each \( j = 0, ..., j^* \), we partition the set of all seeds into \( 2^{j^*-j} \) consecutive groups of size \( 2^j \cdot 47 \), under the size order \( \sigma \), and we call these the large groups. Clearly, each large group of size exceeding 47 is the union of two large groups half its size.

We break up each group \( B \) of size 47 into 12 consecutive groups (consecutive under \( \sigma \)) of the following sizes:
We call these the small groups. (So the small subgroup of size 1 of $B$ consists of the very last seed of $B$ in the size order $\sigma$.) We say the second and sixth groups of size four are of type 1 (they are marked in boldface in Equation 5). The remaining groups of size four (i.e., the first, third, fourth, fifth, seventh, eighth and ninth group of size four) will be called type 2.

It would be difficult to embed the seeds in the size order $\sigma$, as this enumeration might not be suitable for embedding the tree in a connected way. For this reason, we employ a second order $\tau$, which we call the transversal order, obtained by performing a preorder transversal on $T$, starting with the root $r$, and then restricting this order to $W$. (The transversal order is the actual order the seeds will be embedded in.)

The third order, which we call the rearranged order $\rho$, is obtained by reordering the order $\sigma$. First, we reorder the seeds in each small group so that each small group is ordered according to $\tau$. Next, for every large group $B$ of size 47, we reorder all its subgroups so that their first seeds form an increasing sequence in the transversal order $\tau$. Finally, for every large group $B$ of size $>47$ (in successive steps according to the group size), we reorder the two subgroups within $B$ so that the first subgroup contains the first seed in $B$ under the transversal order $\tau$ (i.e., we reorder them such that the first seed of $B$ under $\tau$ becomes the first seed of $B$). This finishes the definition of the rearranged order $\rho$.

We note that $\rho$ maintains the structure given by breaking down the set of seeds into large and small groups. That is, if we partition the set of all seeds into $2^{2^j}$ consecutive groups under $\rho$ of sizes $2^j \cdot 47$, we obtain the same groups as above for $\sigma$. Further, each group $B$ of size 47 breaks down into 12 small groups as above, although these are no longer ordered as in the sequence from (5). (For instance, in $\rho$, the small group of size 1 from $B$ could become the first group in $B$, or be at any other position.)

We will embed according to $\tau$ but momentarily work with $\rho$. We write $s <_\rho s'$ to denote that $s$ comes before $s'$ in order $\rho$ (and similar for $\tau$).

### 5.2.2 Sequences

In this subsection, we will follow the rearranged order $\rho$. We define for each large group $B$ two sequences

$$\left(x^B_i\right)_{i=1,...,j+6} \text{ and } \left(y^B_i\right)_{i=1,...,j+7},$$

where $j$ is such that $|B| = 2^j \cdot 47$, and vertices $x^B_i, y^B_i \in B$ are as specified in what follows.

We construct our sequences inductively. For $j = 0$, we have $2^j$ large groups of size 47. For each such large group $B$, we take $x^B_1 = y^B_1$ as the first seed of the group. The first seed of the second, third, fourth, fifth and sixth small subgroup of $B$ is chosen as $x^B_2, x^B_3, x^B_4, x^B_5, x^B_6$, respectively. The first seed of the seventh, eighth, ninth, tenth, eleventh and twelfth small subgroup of $B$ is chosen as $y^B_2, y^B_3, y^B_4, y^B_5, y^B_6, y^B_7$, respectively.
(We always work under $\rho$, both when talking about the “first seed of a group” and when talking about the “ith subgroup.”)

For $j \geq 1$, we have to deal with all large groups of size $2^j \cdot 47$. For each such group $B$, do the following. By construction $B$ is made up of two large subgroups of size $47 \cdot 2^{j-1}$, say these are $B'$ and $B''$ (in this order, under $\rho$). We set

$$x_i^B := y_i^{B'} \quad \text{for all } 1 \leq i \leq j + 6,$$

and

$$y_i^B := x_i^B = y_i^{B'} \quad \text{and} \quad y_i^B := y_{i-1}^{B''} \quad \text{for all } 2 \leq i \leq j + 7.$$

This finishes the definition of the sequences. We remark that we will only use the sequences $(x_i)$ in what follows (the sequences $(y_i)$ were only used to make the definition of $(x_i)$ more convenient).

Observe that for all blocks $B$, and for all $i < j$, we have that $x_i^B <_\tau x_j^B$.

### 5.2.3 | Relevant seeds

To be later able to choose well the clusters we embed the seeds into (which in turn will enable us to embed the leaves at an even later stage), we need to define, for each seed $s$, a set $X_s$ of relevant seeds for $s$, as follows.

**Definition 5.2** (Relevant seeds for $s$). Let $s$ be a seed of $T$, and let $B$ be the small group $s$ belongs to.

(a) If $B$ is a group of four of type 2, and $s$ is the last seed of $B$, then we set

$$X_s := \{ x : x \text{ is the third seed in } B(\text{under } \rho) \}.$$

(b) If $s$ is not the first seed of $B$, and, in case $B$ is a group of four of type 2, $s$ is not its last seed, then we set

$$X_s := \{ x : x \in B, x < \rho s \}.$$

(c) If $s$ is the first seed of $B$, then we set

$$X_s := \{ x : \exists \tilde{B}, i, i' \text{ such that } i' < i, s = x_i^{\tilde{B}} \text{ and } x = x_{i'}^{\tilde{B}} \}.$$  

Observe that $\tilde{B}$, if it exists, is unique, and that if $s$ only appears as a first vertex in the sequences $(x_i^{\tilde{B}})$, then $X_s = \emptyset$.

Let us make a quick observation which follows directly from the definition of the order $\rho$, of the sequences $(x_i)$ and of the sets $X_s$. 
Observation 5.3. Let $s$ be a seed. Then for all $x \in X_s$ it holds that $x <_r s$.

6 | THE PROOF OF LEMMA 2.1

6.1 | Preparations

6.1.1 | Setting up the constants

First of all, given $\delta$, we choose

$$\varepsilon \leq \frac{\delta^4}{10^{18}},$$

and apply Lemma 4.4 (the regularity lemma) with input $\varepsilon^2$ and $M_0 := \frac{1}{\varepsilon^2}$. This yields numbers $M_1$ and $n_0$. We then set

$$\beta := \frac{\varepsilon}{100M_1}.$$  

Finally, we choose

$$m_\delta := (n_0 + 1) \cdot \frac{400M_0}{\beta^{10} \cdot \varepsilon \cdot \delta},$$

for the output of Lemma 2.1. So, given the approximation constant $\alpha$, satisfying $1 \geq \alpha \geq \delta$, we will have that

$$0 < \frac{1}{m_\delta} \ll \beta \ll \varepsilon \ll \delta \leq \alpha,$$

with the explicit dependencies given above. Now, given $m \geq m_\delta$, and given an $(m + 1)$-vertex graph $G$ of minimum degree at least $\left\lceil \frac{2m}{3} \right\rceil$, and a tree $T$ with at most $m - \alpha m$ edges, rooted at $r$, we will prepare both $T$ and $G$ for the embedding.

6.1.2 | Preparing $T$ for the embedding

We apply Lemma 5.1 to obtain a partition of $T$ into a set $W$ of seeds and a set $\mathcal{T}$ of small trees. The small trees divide into $L$, $F_1$ and $F_2$, with two-seeded trees $F'_2 \subseteq F_2$, and the lemma also gives us a set $\tilde{V}$. Set

$$f_1 := \sum_{T \in F_1} |V(T)| \quad \text{and} \quad f_2 := \sum_{T \in F_2} |V(T)|.$$  

Next, add a set $W'$ of vertices to $T$, each adjacent to $r$, such that, setting $\tilde{W} := W \cup W'$, we have
\[ |\tilde{W}| = 47 \cdot 2^{j^*} \]

for

\[ j^* := \left\lfloor \log \frac{2}{\beta^2} \right\rfloor. \tag{9} \]

The only reason for this is that we plan to apply the grouping and ordering of seeds from Section 5.2, that is, we would like to see (4) fulfilled. We are going to embed \( T \cup W' \) instead of \( T \). Since the number of vertices in \( W' \) is a constant, space is not a problem. Indeed, clearly,

\[ |\tilde{W}| + |L| + f_1 + f_2 = |V(T)| + |\tilde{W} \setminus W| \leq m - \alpha m + \frac{200}{\beta^2}. \tag{10} \]

### 6.1.3 Preparing \( G \) for the embedding

As a preparation of \( G \) for the embedding, we take an \((\varepsilon^2/2)\)-regular partition of \( G \) as given by Lemma 4.4 (the regularity lemma), into \( p \) clusters, for some \( p \) with \( M_0 < p < M_1 \). Consider the reduced graph \( R_G \) of \( G \) with respect to this partition, as defined by Lemma 4.4.

Because of the minimum degree of \( G \) and by (2), we have that

\[ \delta_w(R_G) \geq \left( \frac{2}{3} - 13\varepsilon \right) p. \tag{11} \]

Let us now partition the clusters of \( R_G \) further. We will divide each cluster into several slices, into which we plan to embed the distinct parts of the tree \( T \) which we identified above.

First of all, we choose a set \( Z \) of vertices into which we plan to embed \( L \). More precisely, we arbitrarily choose a set \( Z \subseteq V(G) \) of size

\[ |Z| = |L| + \left\lfloor \alpha - \frac{\alpha^4}{10^6} \right\rfloor m, \tag{12} \]

choosing the same number of vertices in each part of the partition (plus/minus one vertex). Now, we will split up the remainder \( C \setminus Z \) of each cluster \( C \in V(R_G) \) arbitrarily into four sets \( C_{\tilde{V}}, C_W, C_{F_1}, C_{F_2} \), and a leftover set \( C \setminus (Z \cup C_{\tilde{V}} \cup C_W \cup C_{F_1} \cup C_{F_2}) \) which will not be used. The sets are chosen having the following sizes:

\[ |C_{\tilde{V}}| = |C_W| = \left\lceil \frac{\alpha^4 m}{5p} \right\rceil, \tag{13} \]

\[ C_{F_1} = \left\lfloor \frac{f_1 + \alpha^4 m}{5p} \right\rfloor. \tag{14} \]
and

\[ |C_{F_2}| = \left\lfloor \frac{f_2 + \frac{\alpha^4}{\varepsilon} m}{p} \right\rfloor. \]  

(15)

This is possible because of (8) and (10). As we mentioned above, the idea behind this slicing up is that we are planning to put each part \( X \) of the tree \( (X \in \{W, \tilde{V}, F_1, F_2, L\}) \) into the parts \( C_X \) of the clusters of \( R_G \), or into \( Z \), respectively. We reserve a bit more than is actually needed for the embedding, to always be able to choose well-behaved (typical) vertices, and also to account for slightly unbalanced use of the regular pairs when embedding the trees from \( T \). Since the sets \( C_X \) are large enough, regularity properties will be preserved between these sets (cf. Section 4.3).

Let us remark that it is not really necessary to slice the clusters \( C \) up as much as we do: the vertices destined to go into slices \( C_{\tilde{V}} \) and \( C_W \) are actually so few that they could go to any other slice without a problem. But we think the exposition might be clearer if everything is well-controlled.

Finally, we fix a perfect matching \( M_{F_2} \) of \( R_G \) which exists because of (11). This matching will be used for embedding the trees from \( F_2 \).

6.1.4 | The plan

For convenience, for each seed \( s \in \tilde{W} \), let \( T_s \) denote the set of all trees from \( T \setminus L \) that hanging from \( s \). We are going to traverse the seeds in the transversal order \( \tau \), placing each seed \( s \) into a suitable cluster \( S(s) \) (we will determine this cluster right before embedding \( s \) into it). Then embed \( \bigcup T_s \) before embedding any other seed. After having embedded all seeds \( s \in \tilde{W} \) and all corresponding trees from \( \bigcup T_s \), we embed all of \( L \) in one step at the very end of the embedding process. So, if the seeds are ordered as \( s_1, s_2, s_3, \ldots, s_{|\tilde{W}|} \) in \( \tau \), then we embed in the order

\[ s_1, \bigcup T_{s_1}, s_2, \bigcup T_{s_2}, s_3, \bigcup T_{s_3}, \ldots, s_{|\tilde{W}|}, \bigcup T_{s_{|\tilde{W}|}}, L, \]

and at every point in time, the embedded parts of the tree will form a connected set in \( T \).

Each of the three different embedding procedures will be described in detail in one of the following subsections, namely, in Section 6.2 (embedding a seed \( s \)), in Section 6.3 (embedding \( \bigcup T_s \)) and in Section 6.4 (embedding \( L \)).

6.2 | Embedding the seeds

6.2.1 | Preliminaries

Assume we are about to embed some seeds \( s \). Denote by \( U \) the set of vertices that, up to this point, have been used for embedding seeds and small trees. So \( U \cap Z = \emptyset \) (we will ensure that this will always remain so), and every cluster \( C \in V(R_G) \) divides into six sets: \( C \cap U, C \cap Z, C_W \setminus U, C_{\tilde{V}} \setminus U, C_{F_1} \setminus U \) and \( C_{F_2} \setminus U \).
Apart from $U$, it will be useful to have a set $U' \subseteq \bigcup_{C \in V(R_G)} (C_{F_1} \setminus U)$ of vertices for which at some point we decided that they will never be used for the embedding. The main purpose of this set $U'$ is that after embedding certain trees from $T_s \cap F_1$ for some seed $s$, we can just make all sets $C_{F_i}$ of clusters $C$ equally "occupied" by discarding some of the vertices of the emptier sets $C_{F_i}$ by putting them into $U'$. This will be the only time we add vertices to $U'$. We will make sure that for each seed $s$ the number $u'_s$ of vertices we add to $U'$ while, or directly after, embedding $T_s$ is bounded by

$$u'_s \leq \frac{3}{\beta^10} + 800 \varepsilon \cdot |T_s \cap F_1|.$$  

(16)

Since there at most $\frac{2}{\beta}$ (original) seeds in the tree, this means that at any time,

$$|U'| \leq \frac{6}{\beta^{11}} + 800 \varepsilon \cdot \sum_{s \in W} |T_s \cap F_1| \leq 801 \varepsilon m.$$

In other words, the set $U'$ will always stay so small that we can ignore it while embedding.

Throughout the embedding, we will ensure that for each parent $u$ of a seed (the parent $u$ might be a seed, or a vertex from $V_\tilde{G}$) the following holds. If $u$ was embedded in vertex $\phi(u)$, then we have that

$$\psi(u)$$

is typical to slice $C_\psi$ for all but at most $\varepsilon p$ clusters $C$ of $R_G$.

(17)

Note that by Observation 5.3, by the time we reach a seed $s$, the "relevant" seeds in $X_s$ have already been embedded into a set $\phi(X_s)$. Let $N_Z(X_s)$ denote the set of all neighbours of vertices from $X_s$ in $Z$, that is, $N_Z(X_s) := N(X_s) \cap Z$. Let $N_s'$ be the set of the corresponding subsets of the clusters of $R_G$ (i.e., $\bigcup N_s' = N_Z(X_s)$).

6.2.2 Finding the target cluster $S(s)$ for $s$

Before actually choosing the vertex $\phi(s)$ we will embed $s$ into, we will determine the target cluster $S(s)$ for a seed $s$.

Observe that by Lemma 4.1, with $\psi := 13\varepsilon$, we know that at least $\left(\frac{1}{3} + \varepsilon^{\frac{1}{2}}\right)m$ of the vertices of $G$ see a $\left(\frac{1}{2} - \varepsilon^{\frac{1}{2}}\right)$-portion of the vertices in $Z \setminus N_Z(X_s)$. So, for significantly more than a third of the clusters of $R_G$ we have that a significant portion of their vertices see at least $\left(\frac{1}{2} - \varepsilon^{\frac{1}{2}}\right) \cdot |Z \setminus N_Z(X_s)|$ vertices in $Z \setminus N_Z(X_s)$. Because of regularity, and because of (3), this means that for any such cluster $C$, all but at most an $\varepsilon$-fraction of the vertices in $C_\psi$ has at least $\left(\frac{1}{2} - 3\varepsilon^{\frac{1}{2}}\right) \cdot |Z \setminus N_Z(X_s)|$ neighbours in $Z \setminus N_Z(X_s)$.

Choose $S(s)$ as any one of the clusters as above, that is, such that

(\alpha) all but at most $\varepsilon |S(s)_w|$ vertices of the set $S(s)_w$ have degree at least $\left(\frac{1}{2} - 3\varepsilon^{\frac{1}{2}}\right) \cdot |Z \setminus N_Z(X_s)|$ into $Z \setminus N_Z(X_s)$;
and such that in addition (unless \( s = r \), in which case the following two conditions are void),

1. \( S(s) \) is adjacent to \( S(p(s)) \);
2. \( \varphi(p(s)) \) is typical with respect to \( S(s)_{\psi} \),

where \( S(p(s)) \) denotes the cluster the parent \( p(s) \) of \( s \) was embedded into. Such a choice of \( S(s) \) is possible since by (11), cluster \( S(p(s)) \) has a degree almost \( \frac{2n}{3} \) in \( R_G \), and because of (17).

6.2.3 Embedding seed \( s \) into target cluster \( S(s) \)

We place \( s \) in a vertex \( \varphi(s) \) from \( S(s)_{\psi} \setminus U \) such that

1. \( \varphi(s) \) is a neighbour of \( \varphi(p(s)) \) (where \( p(s) \) is the parent of \( s \), and if \( s = r \) this restriction is empty);
2. \( \varphi(s) \) is typical to \( C_{\psi} \) for all but at most \( \varepsilon p \) clusters \( C \in V(R_G) \setminus S(s) \);
3. \( \varphi(s) \) is typical to \( C_{\psi} \) for all but at most \( \varepsilon p \) clusters \( C \in V(R_G) \setminus S(s) \);
4. \( \varphi(s) \) is typical to \( C_{\psi} \setminus (U \cup U') \) for all but at most \( \varepsilon p \) clusters \( C \in V(R_G) \setminus S(s) \) and
5. \( \varphi(s) \) is typical to \( C_L \) for all but at most \( \varepsilon p \) clusters \( C \in V(R_G) \setminus S(s) \).

Such a choice is possible since by (3), almost all vertices in any given cluster are typical with respect to any fixed significant subsets of almost all other clusters. Note that in particular, (E) implies that

\[
\deg_{\varphi}(s) \geq \left( \frac{2}{3} - \varepsilon^2 \right) |Z|.
\] (18)

6.2.4 Degrees of the embedded seeds into \( Z \)

The reason for our choice of \( S(s) \) as a cluster fulfilling property (\( \alpha \)) from Section 6.2.2 is that it allows us to accumulate degree into \( Z \). More precisely, if we consider a seed \( s \) together with its set of relevant seeds \( X_s \), then we know that the union of their neighbourhoods in \( Z \) is significantly larger than the neighbourhood of \( s \) alone. Better still, the more vertices \( X_s \) contains, the larger becomes our bound on this neighbourhood.

Let us make this informal observation more precise in the following claim.

**Claim 6.1.** Let \( B \) be a group of seeds.

1. If \( B \) has size five, then

\[
|N(\varphi(B)) \cap Z| \geq \left( \frac{47}{48} - \varepsilon^2 \right) \cdot |Z|.
\]

2. If \( B \) has size four and is of type 1, then
\(|N(\varphi(B)) \cap Z| \geq \left(\frac{23}{24} - \varepsilon^\frac{1}{2}\right) \cdot |Z|.

(iii) If \(B = \{b_1, b_2, b_3, b_4\}\) (with the seeds \(b_i\) appearing in this order in \(\sigma\)) is of type 2, then

\(|N(\varphi(B)) \cap Z| \geq \left(\frac{11}{12} - \varepsilon^\frac{1}{2}\right) \cdot |Z|

and

\[\min\{|N(\varphi([b_1, b_2])) \cap Z|, |N(\varphi([b_3, b_4])) \cap Z|\} \geq \left(\frac{5}{6} - \varepsilon^\frac{1}{2}\right) \cdot |Z|\]

(iv) If \(B\) is large, say of size \(47 \cdot 2^j\), then

\[|N\left(\varphi\left(\{x_i^B: i = 1,..., j + 6\}\right)\right) \cap Z| \geq \left(1 - \frac{1}{96 \cdot 2^j} - \varepsilon^\frac{1}{2}\right) \cdot |Z|,

where \((x_i^B)_{i=1...j+6}\) is the sequence defined in Section 5.2.2.

Proof. This follows rather directly from (a) and (E) (from Sections 6.2.2 and 6.2.3, respectively), from (18), and from the definition of the set \(X_s\) of relevant seeds (Definition 5.2). For instance, we can calculate the bound in item (i) by using (18), (a), (E), and Definition 5.2(b) to see that

\[|N(\varphi(B)) \cap Z| \geq \left(\frac{2}{3} + \frac{1}{6} + \frac{1}{12} + \frac{1}{24} + \frac{1}{48} - 5 \cdot 3\varepsilon^\frac{1}{2}\right) \cdot |Z| \geq \left(\frac{47}{48} - \varepsilon^\frac{1}{2}\right) \cdot |Z|.

For item (iv), we need to take slightly more care with the calculation. Note that the degree into \(Z\) of the image of the first seed is \(\frac{1}{2}|Z|\) by at most \(3\varepsilon^\frac{1}{2}|Z|\). The degree of second seed's image is only off \(\frac{1}{2}|Z\setminus N_Z(x_1^B)|\) by less than \(3\varepsilon^\frac{1}{2}|Z|\). For the third seed we are only off \(\frac{1}{2}|Z\setminus (N_Z(x_1^B) \cup N_Z(x_2^B))|\) by \(3\varepsilon^\frac{1}{2}|Z|\), and so on, which means we can bound the error in our estimate for the size of the joint neighbourhood in (iv) by

\[\left(1 + \frac{1}{2} + \frac{1}{4} + \cdots\right) \cdot 3\varepsilon^\frac{1}{2}|Z| \leq 2 \cdot 3\varepsilon^\frac{1}{2}|Z| \leq \varepsilon^\frac{1}{4} \cdot |Z|.

6.3 | Embedding the small trees

Assume we have successfully embedded a seed \(s\), and are now, before we proceed to the next seed, about to embed all small trees from \(T_s\) (while still leaving any leaves from \(L\) adjacent to \(s\) unembedded).
Our plan is to embed those trees of $T_s$ that belong to $F_1$ into $\bigcup_{C \in V(R_G)} C_{F_1}$, and those trees of $T_s$ that belong to $F_2$ into $\bigcup_{C \in V(R_G)} C_{F_2}$. We first explain how we deal with the larger trees, that is, those in $F_2 \setminus F'_2$, and those in $F'_2$. After that we explain how we deal with the constant-sized trees, that is, those in $F_1$. Note that actually, it does not matter in which order we deal with the sets $F_1, F_2 \setminus F'_2, F'_2$.

6.3.1 Embedding the trees from $F_2 \setminus F'_2$

For each $T \in \mathcal{T}_s \cap (F_2 \setminus F'_2)$, let $r_T$ denote its root. We plan to put $r_T$ into $C' \setminus U$ for some suitable cluster $C'$. (We will explain below how exactly we do that.) For the rest of $V(T)$, we proceed as follows.

Recall that we defined a perfect matching of $R_G$ near the end of Section 6.1.1. Choose an edge $CD$ of $M_{F_2}$ that contains at least $3\varepsilon \cdot \frac{m}{p}$ unused vertices in each of $C_{F_2}, D_{F_2}$. If $|C_{F_2} \setminus U| \geq |D_{F_2} \setminus U|$, we will aim at putting the larger colour class of $T - r_T$ into $C_{F_2} \setminus U$, and otherwise, we aim at putting it into $D_{F_2} \setminus U$. Observe that if we manage to do this for every tree $T$ we embed, we can ensure that throughout the process (even when embedding trees from $T_{s'}$, for some $s' \neq s$), the edges from $M_{F_2}$ keep their free space in a more or less balanced way, that is, for all edges $C'D'$ in $M_{F_2}$,

$$|C_{F_2} \setminus U| \quad \text{and} \quad |D_{F_2} \setminus U| \quad \text{differ by at most} \quad \frac{\beta m}{p}. \quad (19)$$

Let us now explain how we manage to embed $T$ in this way. Assume our aim is to embed the children of $r_T$ into $C_{F_2} \setminus U$, the grandchildren into $D_{F_2} \setminus U$, the grand-grandchildren into $C_{F_2} \setminus U$ and so on. The embedding of $T - r_T$ will be easy using Lemma 4.5 once we found a vertex $\varphi(r_T)$ to embed $r_T$ into, that is, a vertex that is both a neighbour of $\varphi(s)$ and typical with respect to $C' \setminus U$.

So we only need to find a suitable vertex for $\varphi(r_T)$, the image of the root of $T$ (which belongs to $\hat{V}$). To do so, we first determine a cluster $C'$ that is adjacent to both $C$ and $S(s)$, and that fulfils $d(S(s), C') \geq \frac{1}{4}$. At least nearly a third of the clusters in $R_G$ qualify for this, because of (11). Now, by (C) in the choice of $\varphi(s)$ (in Section 6.2.3), we know that $\varphi(s)$ has a typical degree in the set $C' \setminus U$ for all but very few clusters $C'$. Typical degree means that $\varphi(s)$ has at least $(\frac{1}{4} - \varepsilon^2) \cdot |C' \setminus U|$ vertices in $C' \setminus U$, and by Lemma 5.1(i), at most $2\beta m$ vertices have been used for earlier vertices from $\hat{V}$. So, by (8), we can choose a suitable $C'$ such that $\varphi(s)$ has a large enough neighbourhood in $C' \setminus U$ to ensure it contains a vertex $\varphi(r_T)$ that is typical with respect to $C_{F_2} \setminus U$, that is, such that $\varphi(r_T)$ has at least $(10\sqrt{\varepsilon} - \varepsilon^2) |C_{F_2} \setminus U| \geq \beta m$ neighbours in $C_{F_2} \setminus U$, where the inequality follows from (15) and (19).

6.3.2 Embedding the trees from $F'_2$

For each $T \in \mathcal{T}_s \cap F'_2$, we proceed exactly as in the preceding paragraph, except that now, we have to make a small adjustment when we are close to embedding $\hat{v}$, the second vertex from $\hat{V}$ contained in $V(T)$. 

758 | REED AND STEIN
Suppose \( s' \) is the seed which is adjacent to \( v \) in \( T \). Because we embed the seeds following the transversal order, we know that \( s' \) is not yet embedded by the time we deal with \( T \). We take care to embed \( v \) into a vertex that is typical with respect to almost all the sets \( C_{\tilde{v}} \). That is, the image of \( \tilde{v} \) will be chosen such that (17) holds.

Finally, observe that (15) and (19) ensure that the space we had assigned to \( F_2 \) is enough for embedding first all of \( F_2 \setminus F'_2 \), and now, all of \( F'_2 \).

### 6.3.3 Embedding the trees from \( F_1 \)

We now explain how we embed the trees from \( T_s \cap F_1 \). Note that because of (14) and (16), we have enough space to embed all of \( T_s \cap F_1 \). Furthermore, because the trees from \( F_1 \) are small, and because of regularity, we have no problem with the actual embedding of them into the regular pairs of \( G \). The only thing we need to make sure that the roots of the trees from \( T_s \cap F_1 \) are embedded into neighbours of \( \varphi(s) \), and that we maintain the unused parts of the cluster slices \( C_{F_1} \) balanced at all times.

Since there is no matching like \( M_{F_2} \) that can be used throughout the whole embedding (i.e., for all seeds), we will have to simultaneously keep all of the clusters \( C \) reasonably balanced. This will be possible because of the rather delicate embedding strategy we employ, and which we will start to explain now.

#### Preparing the slices \( C_{F_1} \)

Assume we are about to start the embedding process of the trees from \( T_s \cap F_1 \). First of all, note that we can partition the free space \( \setminus (U \cup \tilde{U}) \) of the slices \( C_{F_1} \) of each of the clusters \( C \in V(R_G) \setminus \{S(s)\} \) into sets \( Q^1, \ldots, Q^r \) for some \( r \), such that \( |Q^i| < 2 \frac{m}{p} \) and \( |Q^i| = \frac{m}{p} \) for \( i = 1, \ldots, r \), and such that for each \( i = 1, \ldots, r \), either all or none of the vertices in \( Q^i \) are adjacent to \( \varphi(s) \). The reason for partitioning the free space in this way is that we now have total control over where exactly the neighbours of \( \varphi(s) \) are (since the sets \( Q^i \) are small enough to be ignored during this step of the embedding). Observe that the sets \( Q^i \), for \( i = 1, \ldots, r \), are still large enough to preserve the regularity properties, although now, in view of (R2) from Section 4.3, we have to replace the regularity parameter \( \frac{\varepsilon^2}{2} \) with \( \frac{\varepsilon^2}{2} + \frac{\varepsilon}{2} = \varepsilon \).

Consider the graph \( H \) with vertex set \( \{Q^i\}_{i=1, \ldots, r, c \in V(R_G)} \) and an edge for each \( \varepsilon \)-regular pair of sufficient density. Say \( H \) has \( p' \) vertices. By (11), the weighted minimum degree of \( R_G \) is bounded by \( \delta_w(R_G) \geq \left( \frac{2}{3} - 13\varepsilon \right)p \), and therefore, the weighted minimum degree of \( H \) is bounded by \( \delta_w(H) \geq \left( \frac{2}{3} - 17\varepsilon \right)p' \).

So, by our choice of \( \varphi(s) \), in particular by (D) of Section 6.2.3, we know that \( \varphi(s) \) is typical to almost all clusters, and thus has neighbours in at least \( \left( \frac{2}{3} - 20\varepsilon \right)p' \) of the sets \( Q^i \). Let \( N \) consist of a set of \( \left| \left( \frac{2}{3} - 20\varepsilon \right)p' \right| \) sets \( Q^i \) that contain neighbours of \( \varphi(s) \). We apply Lemma 4.3 with \( \xi := 17\varepsilon \) to \( H \) to obtain a set \( X \), of size

\[
\|X\| \leq \left| 391\varepsilon p' \right| + 3 \leq 400\varepsilon p',
\]
as well as an \((N \setminus X)\)-good matching \(M (\equiv M_t)\), an \((N \setminus X)\)-in-good path partition \(\mathcal{P}_A\) and an \((N \setminus X)\)-out-good path partition \(\mathcal{P}_B\) of \(H - X\).

Set \(Q := \bigcup_{C \in \mathcal{V}(R_C)} \{Q_C^1, Q_C^2, ..., Q_C^c\} \setminus X\). By (14), and by (16), we know that \(\bigcup Q\) is large enough to host all of \(\bigcup \mathcal{T}_s \cap F_1\). In fact, if \(\bigcup \mathcal{T}_s \cap F_1\) could be embedded absolutely balanced into the sets \(Q \in Q\), then there would even be a leftover space of more than \(100e^m\) in each of the sets \(Q\).

Recall that during the embedding of the trees from \(\mathcal{T}_s \cap F_1\), we will add some vertices to a set \(U^\prime\), for keeping better track of the balancing of the edges. We will keep \(U^\prime\) small, that is, we will ensure that (16) holds.

### Preparing \(\mathcal{T}_s \cap F_1\)

We now partition the set of trees from \(\mathcal{T}_s \cap F_1\) into three sets: the set \(T_{\text{Bal}}\) contains all the balanced trees, that is, those trees whose colour classes have the same size; the set \(T_{\text{NearBal}}\) contains all trees having the property that their colour classes differ by exactly one, with the bigger class containing the root; and the set \(T_{\text{Unbal}}\) contains all the remaining trees, that is all unbalanced trees not belonging to \(T_{\text{NearBal}}\).

#### Phase 1

In the first phase of our embedding, we embed all trees from \(T_{\text{Bal}}\), using the matching \(M\). We try to spread these trees as evenly as possible among the edges of \(M\). It is not difficult to see that by Lemma 5.1(e), it is possible to make the used part of the clusters differ by at most \(\beta^1\) (but even the more obvious weaker bound \(\frac{1}{\beta^1}\) is sufficient for our purposes). At the end of this phase of the embedding, we add to \(U^\prime\) at most \(\frac{1}{\beta^1}\) unused vertices from each of the clusters \(Q \in \mathcal{V}(M)\), and can thus make sure each of the clusters has exactly the same number of vertices in \(\bigcup Q\).

#### Phase 2

In the second phase of our embedding, we embed all trees from \(T_{\text{Unbal}}\). We group the trees from \(T_{\text{Unbal}}\) by their number of vertices, which is some number between 3 and \(\frac{1}{\beta^1}\). Then we subdivide these groups according to the number of vertices belonging to the same colour class as their root. The final groups represent the types of trees. Since all trees we consider have to order at most \(\frac{1}{\beta^2}\),

\[
\text{there are at most } \frac{1}{\beta^2} \text{ different types of trees in } T_{\text{Unbal}}. \tag{21}
\]

For each of the types \(T\), say with \(t\) vertices, and colour classes of sizes \(t_1\) and \(t_2\), where the class of size \(t_1\) contains the root, we proceed as follows. We go through the elements of our \(N\)-out-good path partition \(\mathcal{P}_B\) in some fixed order, always embedding only a constant number of trees of type \(T\). In each round, we keep the clusters of \(H - X\) perfectly balanced. Only when we run out of trees of type \(T\), we will (necessarily) have to make a last round, possibly not reaching all elements of \(\mathcal{P}_B\), and thus unbalancing some of the clusters a bit (by at most \(\frac{1}{\beta^1}\)).

---

3We remark that it is not really necessary to treat the trees from \(T_{\text{Bal}}\) separately (as they could be treated together with the trees from \(T_{\text{Unbal}}\) in Phase 2), but we believe that embedding \(\bigcup T_{\text{Bal}}\) first (in Phase 1) is more instructive.
To make the above description more precise, recall that $\mathcal{P}_B$ consists of

(M1) single edges $AB$ with both ends in $N$;
(M2) paths $ABCD$ with $B, C \in N$ and
(M3) paths $ABCDEF$ with $B, C, D, E \in N$.

Say there are $m_1$ paths $AB$ as in (M1), $m_2$ paths $ABCD$ as in (M2), and $m_3$ paths $ABCDEF$ as in (M3). Let us now analyse how the sets $Q \in Q$ lying in edges or paths from (M1)–(M3) fill up when we embed small trees of type $T$ into them in the following specific ways.

**Paths as in (M1)**
First, the sets $AB$, of any edge as in (M1) will each get filled up with $t$ vertices if we embed one tree of type $T$ in one “direction” and a second tree of type $T$ in the other “direction.” In other words, we can embed a total of $2m_1h$ trees of type $T$ into the edges from (M1), filling each of the corresponding clusters $Q$ with $th$ vertices (where $h$ is any not too large natural number).

**Paths as in (M2)**
The sets $Q$ on paths $ABCD$ as in (M2) will get filled as follows. If we

- perform $x$ rounds in which we embed one tree of the current type $T$ in the edge $AB$, with the root going to $B$;
- perform $x$ rounds in which we embed a tree of type $T$ in the edge $CD$, with the root going to $C$;
- perform $y$ rounds of embedding a tree of type $T$ with the root going to $C$, but the rest of the tree going to $AB$ and
- perform $y$ rounds of embedding a tree of type $T$ with the root going to $B$, but the rest of the tree going to $CD$,

then after these $2x + 2y$ rounds, $A$ and $D$ each have received $xt_2 + y(t_1 - 1)$ vertices, while $B$ and $C$ each have received $xt_1 + y(t_2 + 1)$ vertices.

So, if $t_1 > t_2$ (observe that then actually $t_1 \geq t_2 + 2$, since $T \notin T_{\text{NearBal}}$), we will have filled each of the four sets $A, B, C, D$ with exactly $(t_1 - t_2 - 1)t$ vertices if we choose $x = t_1 - t_2 - 2$ and $y = t_1 - t_2$. If $t_2 \geq t_1$, we can fill each of the four sets $A, B, C, D$ with exactly $(t_2 - t_1 + 1)t$ vertices by taking $x = t_2 - t_1 + 2$ and $y = t_2 - t_1$.

Resumingly, for any not too large natural number $h'$, there is a way to embed a total of $|t_1 - t_2 - 1| \cdot 4m_2h'$ trees of type $T$ into the edges from (M2), filling up each of the corresponding clusters $Q$ with $|t_1 - t_2 - 1| \cdot t$ vertices. Even more, taking into account what we said above for edges from (M1), we conclude that for any not too large $h'$, we are able to embed a total of

$$|t_1 - t_2 - 1| \cdot 2m_1h' + |t_1 - t_2 - 1| \cdot 4m_2h' = |t_1 - t_2 - 1| \cdot (2m_1 + 4m_2)h'$$

trees of type $T$ into the edges from (M1) and (M2), filling up each of the corresponding clusters $Q$ with $|t_1 - t_2 - 1| \cdot th'$ vertices.
**Paths as in (M3)**

For the paths ABCDEF from (M3) we can calculate similarly: Say we do \( x \) rounds of embedding of a tree of type \( T \) in the edge \( AB \) and another \( x \) rounds embedding it into \( EF \). We then do \( y \) rounds of embedding the tree into \( AB \), but with the root of the tree going into \( C \), and another \( y \) rounds putting it into \( EF \), with the root going into \( D \). Moreover, we perform \( 2\varepsilon \) rounds where we embed the tree into \( CD \), of which \( \varepsilon \) rounds in each “direction.” Then after these \( 2x + 2y + 2\varepsilon \) rounds, we filled each of \( A \) and \( F \) with \( x_2 + y(t_1 - 1) \) vertices, each of \( B \) and \( E \) with \( x_1 + y_2 \) vertices, and each of \( C \) and \( D \) with \( y + y_1 \) vertices.

So, if \( t_2 \geq t_1 \), then with \( x = t \cdot (t_2 - t_1 + 1), y = t \cdot (t_2 - t_1) \) and \( z = (t - 1) \cdot (t_2 - t_1) + t_1 \), we have filled each of the six sets \( A, B, C, D, E, F \) with exactly the same amount of vertices, namely, with \( t \cdot (t_2 - t_1 + 1) \) vertices each. If \( t_1 > t_2 \), we choose \( x = t \cdot (t_1 - t_2 - 1), y = t \cdot (t_1 - t_2) \) and \( z = (t - 1) \cdot (t_1 - t_2) - t_1 \), and fill each of the six sets \( A, B, C, D, E, F \) with \( t \cdot (t_1 - t_2 - 1) \) vertices. So, adopting the convention that \( \pm t_1 \) means \( +t_1 \) if \( t_2 \geq t_1 \) and \(-t_1\) otherwise, we can embed, for any not too large \( h'' \), a total of \( 6(t \cdot |t_2 - t_1| \pm t_1)m_3h'' \) trees of type \( T \) into the edges from (M3), placing exactly \( t \cdot (t \cdot |t_2 - t_1| \pm t_1)h'' \) vertices into each of the corresponding clusters \( Q \).

Recalling the earlier observations on embedding trees of type \( \overline{T} \) into edges from (M1) and (M2), we conclude that we are able to embed

\[
\begin{align*}
d_T &= |t_1 - t_2 - 1| \cdot (2m_1 + 4m_2)(t \cdot |t_2 - t_1| \pm t_1) \\
&\quad + 6(t \cdot |t_2 - t_1| \pm t_1)m_3 \cdot |t_1 - t_2 - 1|
\end{align*}
\]

trees of the current type \( T \), using all sets \( Q \subseteq Q \) in a completely balanced way (each receives exactly \( t \cdot (t \cdot |t_2 - t_1| \pm t_1) \cdot |t_1 - t_2 - 1| \) vertices).

Now, we embed the first \( d_T \) trees of type \( T \) in this way, then proceed to embed the next \( d_T \) such trees, and so on. If at some point (this might happen in the first round already), there are less than \( d_T \) trees of type \( T \) left, then we perform the last round for embedding these trees, simultaneously blocking at most \( d_T \cdot t \) vertices which we add to \( U' \) (note that they have not been used for the embedding). In this way, we can finish the embedding of all the trees of the current type \( T \) while perfectly balancing \( Q \cap (U \cup U') \) for all clusters \( Q \subseteq Q \). In particular, each of the clusters has exactly the same number of vertices in \( Q \setminus (U \cup U') \).

Note that, when working on one \( t \)-vertex tree \( T \in T_{\text{Unbal}} \), the number of vertices we add to \( U' \) although they are not actually used for the embedding is at most

\[
d_T t \leq 2t^4 \cdot p' \leq \frac{2}{\beta^4} \cdot \frac{p}{\varepsilon} \leq \frac{1}{50\beta^5} \leq \frac{1}{\beta^5},
\]

where we used that each tree in \( T_{\text{Unbal}} \subseteq F_1 \) has at most \( \frac{1}{\beta} \) vertices. So, the number of vertices we add to \( U' \) after working on all trees from \( T_{\text{Unbal}} \) is at most the number of types of trees multiplied by \( \frac{1}{\beta^5} \), and thus, by (21), at most \( \frac{1}{\beta^{10}} \).

**Phase 3**

In this phase, we embed the trees from \( T_{\text{NearBal}} \). Each of these trees has (at least) one leaf in its heavier colour class. Instead of the root, as in phase 2, we will now put this leaf into a different
cluster, and instead of the $N$-out-good path partition we will be using the $N$-in-good path partition $\mathcal{P}_A$.

Again we go through the different types $T$ of trees, of which there are at most $\frac{1}{2\beta}$. Now say we are working on the trees of a fixed type $T$, with $t$ vertices. Easier considerations than in the previous case show that we can embed exactly $t$ vertices into each of the slices $Q_{F_i}$ of clusters $Q \in V(M)$ if we embed six trees of type $T$ into the six clusters corresponding to a path of length 6, four trees of type $T$ into the four clusters corresponding to a path of length 4, and two trees of type $T$ into the clusters corresponding to a path of length 2. So, putting at most $\frac{1}{2\beta} \cdot \frac{1}{\beta} \cdot p' \leq \frac{1}{\beta^3}$ unused vertices into $U'$, we can finish the embedding of all the trees of $T_{\text{NearBal}}$ balancing all slices as desired. In particular, each of the clusters $Q \in \mathcal{Q}$ has exactly the same number of vertices in $Q_{F_i} \setminus (U \cup U')$.

After finishing Phase 3, we still put some more vertices into $U'$, before declaring the embedding procedure of the trees in $\mathcal{F}_s \cap F_1$ finished. Namely, we put an appropriate number of vertices from the sets in $X$ into $U'$. That is, the number of vertices from any of the sets of $X$ we add to $U'$ is the same as the number of vertices from any of the sets $Q \in \mathcal{Q}$ that went to $U$ or to $U'$ during the embedding of $T_s \cap F_1$. This cleaning-up is only done because it will be nicer to be able to start the embedding of the trees at the next seed with all slices $C_{F_1}$ perfectly balanced.

Observe that the number of vertices we added to $U'$ while dealing with the trees from $T_s \cap F_1$ is at most

$$u'_s \leq \frac{1}{\beta} + \frac{1}{\beta^{10}} + \frac{6}{\beta^3} + |X| \cdot \frac{2|T_s \cap F_1|}{p'} \leq \frac{3}{\beta^{10}} + 800 \varepsilon \cdot |T_s \cap F_1|,$$

where we used (7) and (20) for the last inequality. Hence the bound (16) we had claimed above is correct. This ensures we have enough space for all future trees from $F_1$.

### 6.4 Embedding the leaves

This section is devoted to the embedding of the leaves. That is, we are now at a stage where we have successfully embedded all seeds and all small trees, and all that is left to embed is $L$, the set of leaves adjacent to seeds. We will show we can embed all of $L$ at once.

If we cannot embed $L$ into $Z$, then by Hall’s theorem,\(^5\) there is some subset $K \subseteq \hat{W}$ such that

$$|N(\varphi(K)) \cap Z| < |L_K|,$$

where $L_K$ is the set of leaves adjacent to elements of $K$, the set $\varphi(K)$ is the set of images of $K$, and $N(\varphi(K)) \cap Z$ is the union of the neighbours in $Z$ of the elements of $\varphi(K)$.

\(^4\)Note that we did not add $\bigcup_{C \in V(R_0)} (Q_C^L \cup Q_C^R)$ to $U'$.

\(^5\)Hall’s theorem can be found in any standard textbook, it states that a bipartite graph with bipartition classes $A$ and $B$ either has a matching covering all of $A$, or there is an “obstruction”: a set $A' \subseteq A$ such that $|N(A')| < |AA'|$. 
Recall that by (E) from Section 6.2.3, we chose as the image of a seed $s$ a vertex $\varphi(s)$ that is typical with respect to $C_L$ for almost all clusters $C$ of $R_G$. Because of (11), this means that

$$\text{each element of } \varphi(K) \text{ sees at least } \left(\frac{2}{3} - 20\varepsilon\right)|Z| \text{ vertices of } Z.$$  

(23)

In particular, by (12), each element of $\varphi(K)$ sees more than $\frac{5}{8}(|L| + \frac{9}{10}am)$ vertices of $Z$. Thus, we may assume that

$$|L| > \frac{3am}{2},$$

(24)

as otherwise $\frac{5}{8}(|L| + \frac{9}{10}am) \geq |L|$, which means that we could have embedded $L$ without a problem.

Our aim is to reach a contradiction to the assumption that the set $K$ exists. We will reach this contradiction by proving in Claims 6.2–6.6 that $K$ misses a vertex in each of the large groups and also in most of the small groups of seeds we defined in Section 5.2.1. In some of the small groups $K$ actually misses more than one vertex. We will prove these claims by repeatedly using (22). This means that in total, $K$ misses many vertices from $\tilde{W}$, and these vertices spread out among the blocks (and thus have a corresponding proportion of the leaves hanging from them). Therefore, we can conclude that $|L_K|$ is smaller than the bound for the neighbourhood of $\varphi(K)$ given in (23), and thus, $L_K$ could have been embedded without a problem, which is a contradiction.

Let us make this outline more precise. We start by proving that each of the large groups has a vertex outside $K$.

**Claim 6.2.** No large group is completely contained in $K$.

**Proof:** Assume otherwise, and consider the largest $j$ for which there is a group of size $47 \cdot 2^j$ completely contained in $K$. Then by Claim 6.1(iv) from Section 6.2.4, we know that

$$|N(\varphi(K)) \cap Z| \geq \left(1 - \frac{1}{95 \cdot 2^j} - \varepsilon^j\right)|Z|.$$  

(25)

If $j \geq j^\oplus := \left\lceil \log \frac{1}{95 \cdot \frac{am}{1000}} \right\rceil$, then by (12), the bound from (25) exceeds $|L|$, which yields a contradiction to (22). So

$$j < j^\oplus.$$  

(26)

In particular, because of (8) and (9), we know that $j < j^*$, and hence, there exists a large group of size $47 \cdot 2^{j+1} = 94 \cdot 2^j$. For any such group $B$, we know that, by the choice of $j$, there is a vertex $v_B \in B$ that is not in $K$. Let $L_B$ be the set of leaves adjacent to seeds in $B$, and let $\ell_{\text{max}}(B)$ and $\ell_{\text{min}}(B)$ be the number of leaves adjacent to the first and the last
seed in $B$, respectively, under the size order $\sigma$. Thus, every seed $b \in B$ is adjacent to a number $\ell_b$ of leaves, with $\ell_{\text{min}}(B) \leq \ell_b \leq \ell_{\text{max}}(B)$.

Set

\[
\text{dif}(B) := \ell_{\text{max}}(B) - \ell_{\text{min}}(B).
\]

Then

\[
\ell_{vb} \geq \ell_{\text{min}}(B) \geq \frac{|L_b|}{|B|} - \text{dif}(B) \cdot \frac{|B| - 1}{|B|}
= \frac{|L_b|}{94 \cdot 2^j} - \text{dif}(B) \cdot \left(1 - \frac{1}{94 \cdot 2^j}\right).
\]

Since the groups are consecutive in the size order $\sigma$, and since no seed has more than $\alpha m$ leaves adjacent to it, we know that

\[
\sum_{B: |B| = 94 \cdot 2^j} \text{dif}(B) \leq \alpha m.
\]

So, the number of leaves adjacent to seeds that are not in $K$ can be bounded by calculating

\[
|L| - |L_k| \geq \sum_{B: |B| = 94 \cdot 2^j} \ell_{vb}
\geq \frac{|L|}{94 \cdot 2^j} - \sum_{B: |B| = 94 \cdot 2^j} \text{dif}(B) \cdot \left(1 - \frac{1}{94 \cdot 2^j}\right)
\geq \frac{|L|}{94 \cdot 2^j} - \alpha m \cdot \left(1 - \frac{1}{94 \cdot 2^j}\right).
\]

Therefore,

\[
|L_k| \leq \left(1 - \frac{1}{94 \cdot 2^j}\right) \cdot (|L| + \alpha m)
\leq \left(1 - \frac{1}{94 \cdot 2^j}\right) \cdot \left(|Z| + \frac{\alpha^4 m}{10^6}\right)
\leq \left(1 - \frac{1}{95 \cdot 2^j} - \varepsilon^j\right)|Z|
\leq |N(\phi(K)) \cap Z|,
\]

where the second and last inequalities follow from (12) and (25), respectively, and the third inequality follows from the observation that

\[
\frac{\alpha^4 m}{10^6} \leq \frac{2}{3} \cdot \frac{\alpha^3 |Z|}{10^6} \leq \left(\frac{1}{94 \cdot 95 \cdot 2^j} - \varepsilon^j\right)|Z|,
\]
where for the first inequality we used that \(|Z| \geq |L| > \frac{3am}{2}\) (by Equation 24), and the second inequality follows from the facts that \(\varepsilon \leq \frac{\alpha^4}{10^6}\) (by Equation 6) and \(j \leq j^\circ\) (by Equation 26).

Now, inequality (27) gives a contradiction to (22). This proves Claim 6.2.

Next, we show a similar fact for all small groups of size five.

**Claim 6.3.** No small group of size 5 is completely contained in \(K\).

**Proof.** Indeed, otherwise, because of Claim 6.1(i), we know that

\[
|N(\varphi(K)) \cap Z| \geq \left(\frac{47}{48} - \varepsilon^4\right)|Z|.
\]  

(28)

Moreover, Claim 6.2 implies that every large group \(B\) of size 47 has a vertex \(v_B\) which is not in \(K\), and thus we can calculate, similar as above for Claim 6.2, that

\[
|L| - |L_K| \geq \sum_{B: |B|=47} e_{v_B} \geq \frac{|L|}{47} - \alpha m \cdot \left(1 - \frac{1}{47}\right),
\]

and thus, employing (12), we find that

\[
|L_K| \leq \left(1 - \frac{1}{47}\right) \cdot \left(|Z| + \frac{\alpha^4}{10^6} m\right),
\]

which, with the help of (24) and (28), and using the fact that \(\alpha \gg \varepsilon\), yields a contradiction to (22). This proves Claim 6.3.

Next, we turn to the groups of size four, separating the treatment of these into two cases depending on their type.

**Claim 6.4.** No small group of size 4 and of type 1 is completely contained in \(K\).

**Proof.** Otherwise, because of Claim 6.1(ii), we know that

\[
|N(\varphi(K)) \cap Z| \geq \left(\frac{23}{24} - \varepsilon^4\right)|Z|.
\]  

(29)

By Claim 6.3 we know that every group of size five has a vertex which is not in \(K\). Hence, every large group \(B\) of size 47 contains at least two vertices \(v_B^1\) and \(v_B^2\) that are not in \(K\). Moreover, by the definition of the small groups, the first of these vertices, \(v_B^1\), is one of the first 23 vertices of \(B\) under the size order, and the second vertex \(v_B^2\), is one of the next 23 vertices of \(B\) under the size order.

So, we can split the group \(B\) minus its last vertex into two groups \(B_1, B_2\) containing the first 23 and the next 23 consecutive elements in the size order, respectively, with \(v_B^1 \in B_1\) and \(v_B^2 \in B_2\). Defining \(\text{dif}\ (B_1), \text{dif}\ (B_2)\) as in Claim 6.2 for each of these two subgroups
$B_1, B_2$ of the group $B$ of size 47, and letting $\ell_{v_B}^i$ denote the number of leaves hanging from $v_B^i$, for $i = 1, 2$, we can calculate that

$$
\ell_{v_B}^i \geq \frac{|L_B|}{23} - \frac{22}{23} \cdot \text{dif}(B_1)
$$

and

$$
\ell_{v_B}^i \geq \frac{|L_B|}{23} - \frac{22}{23} \cdot \text{dif}(B_2),
$$

where $L_{B_1}$ and $L_{B_2}$ denote the sets of leaves adjacent to vertices from $B_1$ and $B_2$, respectively. Thus, letting $L_{B_3}$ denote the set of leaves adjacent to the very last vertex of the group $B$ (of size 47), and noticing that $|L_{B_3}| \leq \frac{|a|}{47}$, we can calculate that

$$
|L| - |L_K| \geq \sum_{B: |B| = 47} \left( \ell_{v_B}^1 + \ell_{v_B}^2 \right)
$$

$$
\geq \frac{|L|}{23} - \frac{22}{23} \cdot \text{dif}(B_1) \geq \frac{|L|}{23} - \frac{22}{23} \cdot \text{dif}(B_2).
$$

Therefore, using (12) and (29), we obtain that

$$
|L_K| \leq \frac{22}{23} \left( |L| + \alpha m \right) \cdot \left( 1 + \frac{1}{22 \cdot 47} \right)
$$

$$
\leq \frac{22}{23} \left( |Z| + \frac{\alpha}{10^6} m \right) \cdot \left( 1 + \frac{1}{22 \cdot 47} \right)
$$

$$
\leq \left( \frac{23}{24} - \varepsilon \frac{1}{2} \right) |Z|
$$

$$
\leq |N(\varphi(K)) \cap Z|,
$$

a contradiction to (22). This proves Claim 6.4.

**Claim 6.5.** No small group of size 4 and of type 2 is completely contained in $K$.

**Proof.** Otherwise, because of Claim 6.1(iii), we know that

$$
|N(\varphi(K)) \cap Z| \geq \left( \frac{11}{12} - \varepsilon \frac{1}{2} \right) |Z|.
$$

However, by Claims 6.3 and 6.4, we know that every small group of size 5 and every small group of size 4 and of type 1 has a vertex which is not in $K$. So, we can split every large group $B$ of size 47 into five subgroups $B_1, B_2, B_3, B_4, B_5$, each consecutive in the size order,
and with $B_i = 11$ for $i = 1, 2, 3, 4$ and $|B_d| = 3$, such that each of $B_1, B_2, B_3, B_4$ contains a vertex $v^1_B, v^2_B, v^3_B, v^4_B \notin K$.

Similar as the above, we can calculate that

$$|L| - |L_K| \geq \sum_{B: |B| = 47} \left( \ell_{v_B}^1 + \ell_{v_B}^2 + \ell_{v_B}^3 + \ell_{v_B}^4 \right),$$

$$\geq \frac{|L|}{11} - \frac{3|L|}{11 \cdot 47} - \frac{10}{11} \alpha m,$$

where numbers $\ell_{v_B}$ are defined as in the previous claim. Now, using (12) and (30), we obtain that

$$|L_K| \leq \frac{10}{11} (|L| + \alpha m) \cdot \left( 1 + \frac{1}{10 \cdot 15} \right),$$

$$\leq \left( \frac{11}{12} - \frac{1}{12} \right) |Z|,$$

$$\leq |N(\varphi(K)) \cap Z|,$$

a contradiction to (22). This proves Claim 6.5.

Next, we will show that we can actually get some more out of the groups of type 2.

**Claim 6.6.** No small group of size 4 and of type 2 has three or more vertices in $K$.

**Proof.** Indeed, otherwise, because of the second part of Claim 6.1(iii), we know that

$$|N(\varphi(K)) \cap Z| \geq \left( \frac{5}{6} - \varepsilon \right) |Z|.$$  \hspace{1cm} (31)

By Claims 6.3–6.5, every small group $B$, except possibly those of size one, has a vertex $v_B$ which is not in $K$. So, similar as in the previous claims, but now going over all groups of sizes 4 and 5, and temporarily considering the groups of size 1 to form part of the previous group (which had size 4, and so now has size 5), we calculate that

$$|L| - |L_K| \geq \sum_{B: 4 \leq |B| \leq 5} \ell_{v_B} \geq \sum_{B: 4 \leq |B| \leq 5} \left( \frac{|L_B|}{|B|} - \text{dif} (B) \cdot \frac{|B| - 1}{|B|} \right),$$

$$\geq \min \left\{ \frac{1}{4}, \frac{1}{5} \right\} \cdot |L| - \max \left\{ \frac{3}{4}, \frac{4}{5} \right\} \cdot \alpha m,$$

$$\geq \frac{|L|}{5} - \frac{4}{5} \alpha m,$$

where $L_B$ is the set of leaves incident with vertices of $B$, and $\text{dif} (B)$ is the difference between the biggest number of leaves hanging from a vertex in $B$ and the smallest such number. We then use (12) and (31), to obtain that
\[ |L_K| \leq \frac{4}{5} (|L| + \alpha m) \leq \left( \frac{5}{6} - \varepsilon \right) |Z| \leq |N(\varphi(K)) \cap Z|, \]

a contradiction to (22). This proves Claim 6.6. □

Resumingly, Claims 6.3–6.6 tell us that \( K \) misses at least one vertex of each small group of size four or five, and misses at least two vertices from each small group of size four and type 2. Recalling our ordering \( 4, 4, 4, 4, 5, 4, 4, 4, 4, 4, 5, 4, 4, 4, 4, 4, 5, 4 \) of the small groups inside each group \( B \) of size 47 as given in (5) in Section 5.2 (under ordering \( \sigma \)), we see that we can split \( B \) into five groups \( B_1, B_2, B_3, B_4, B_5 \) such that

- for \( i = 1, 3 \), the group \( B_i \) has eight vertices, at least 3 of which are not in \( K \);
- for \( i = 2, 4 \), the group \( B_i \) has 13 vertices, at least 5 of which are not in \( K \) and
- \( B_5 \) has five vertices, at least two of which are not in \( K \).

Therefore, similar calculations as for the previous claim give that

\[
|L| - |L_K| \geq \min \left\{ \frac{3}{8}, \frac{5}{13}, \frac{2}{5} \right\} \cdot |L| - \max \left\{ \frac{8}{13}, \frac{3}{5} \right\} \cdot \alpha m
\]

\[
\geq \frac{3|L|}{8} - \frac{5}{8} \alpha m,
\]

and thus by (12), we get

\[
|L_K| \leq \frac{5}{8} (|L| + \alpha m) < \left( \frac{2}{3} - 20\varepsilon \right) |Z|,
\]

a contradiction to (23). This means the Hall-obstruction \( K \) cannot exist, and we can thus finish the embedding of \( T \) by embedding all leaves from \( L \) in one step. This finishes the proof of Lemma 2.1.

7 | EXTENDING A GIVEN EMBEDDING

For the companion paper [18], which contains the proof of the exact version of Theorem 1.4, we will not only need Lemma 2.1, but also a second result, namely, Lemma 7.3, the main result of this section, which is stated below. Both lemmas are very similar.

The main difference in Lemma 7.3 is that in the context of [18], a small tree \( T^* \) is already embedded, except for a small set \( Y \subseteq V(T^*) \). As images of neighbours of \( Y \) are well chosen, we will later be able to absorb \( Y \), that is, we choose a suitable set \( S \subseteq V(G) \) somewhat smaller than \( Y \), embed \( T - (T^* - Y) \) into \( G - S \), and then complete the embedding by using the leftover plus \( S \) for the embedding of \( Y \). So, in Lemma 7.3, we wish to embed \( T - T^* \), and just as in Lemma 2.1 we have some extra free space, but now we have to cope with the already embedded \( T^* - Y \), which may block neighbourhoods. However, as we will see below, our proof of Lemma 2.1 can be adapted to the new setting, with two possible exceptions.
First, if $G$ is $\gamma$-special (see Definition 7.1), our embedding scheme will fail, because the matchings we need for the embedding of $F_1$ might not exist. In [18] we show how to embed $T$ in that case. Second, if $T$ has a very specific shape, and $G$ is close to containing a complete tripartite graph, we may not be able to find the in-good path partition needed for embedding $T_{\text{NearBal}}$. This case is covered by Lemma 7.5, which shows that then we can embed all of $T$.

Another important difference to Lemma 2.1 is that in Lemma 7.3, we know that no seed of $T$ has many leaves hanging from it. So, we can forget about all the extra work that was done in the proof of Lemma 2.1 to embed the set $L$ of leaves hanging from seeds.

Let us now give the two definitions we need to state Lemma 7.3.

**Definition 7.1.** We say a graph $G$ on $m + 1$ vertices is $\gamma$-special, for some $\gamma > 0$, if $V(G)$ consists of three mutually disjoint sets $X_1, X_2, X_3$ such that

(i) $\frac{m}{3} - 3\gamma m \leq |X_i| \leq \frac{m}{3} + 3\gamma m$ for each $i = 1, 2, 3$ and

(ii) there are at most $\gamma^{10} |X_1| \cdot |X_2|$ edges between $X_1$ and $X_2$.

**Definition 7.2.** Let $T$ be a tree with $m$ edges. Call a subtree $T^*$ of $T$ with root $t^*$ a $\gamma$-nice subtree if $|V(T^*)| < \gamma m$ and every component of $T - T^*$ is adjacent to $t^*$.

We are now ready for the result we will need in [18].

**Lemma 7.3.** For all $\gamma < \frac{1}{10}$ there are $m_0 \in \mathbb{N}$ and $\lambda > 0$ such that the following holds for all $m \geq m_0$.

Let $G$ be an $(m + 1)$-vertex graph of minimum degree at least $\lfloor \frac{2m}{3} \rfloor$ and with a universal vertex, such that $G$ is not $\gamma$-special. Let $T$ be a tree with $m$ edges such that $T \not\subseteq G$ and no vertex in $T$ is adjacent to more than $\lambda m$ leaves. Let $T^*$ be a $\gamma$-nice subtree of $T$ with root $t^*$, let $Y \subseteq V(T^*) \setminus \{t^*\}$, and let $S \subseteq V(G)$ with $|S| \leq |Y| - \left(\frac{\lambda}{2}\right)^4 m$.

Assume that for any $W \subseteq V(G) - S$ with $|W| \geq \gamma m$, there is an embedding $\varphi_W$ of $T^* - Y$ into $G - S$, with $t^*$ embedded in $W$. Then there is a set $W \subseteq V(G) - S$ with $|W| \geq \gamma m$, and an embedding of $T - Y$ into $G - S$ that extends $\varphi_W$.

In the proof of Lemma 7.3, we will need the following definition and lemma.

**Definition 7.4.** Call a rooted tree bad if it is a three-vertex path whose root is not the middle vertex.

**Lemma 7.5.** Let $H$ be a graph on $m + 1$ vertices with universal vertex $w$. Let $H_1 \cup \cdots \cup H_5$ be a partition of $V(H)$, with $|H_1| = |H_2| = |H_3| \geq \frac{33}{100} m$ and $H_4 \leq \frac{1}{5}|H_5|$. Assume that for all $i, j \in \{1, 2, 3\}$ with $i \neq j$, each vertex of $H_i$ is adjacent to at least $\frac{99}{100}$ of the vertices in $H_j$, and for $i = 4, 5$, each vertex of $H_i$ is adjacent to at least two-fifths of the vertices of $H_{i-2}$.
Let \( T \) be a tree with \( m \) edges, and let \( W, L, F_1, F_2 \) be sets as in Lemma 5.1 for some \( \beta \) with \( \beta^2m > 250 \), except that we do not require the upper bound in Lemma 5.1\((f)\). If \( F_1 \) contains at least \( \frac{33}{100}m \) bad trees, then \( T \subseteq G \).

We leave the proof of Lemma 7.5 to the end of the section, and first prove Lemma 7.3, mainly following the lines of the proof of Lemma 2.1.

**Proof of Lemma 7.3.** We structure the proof according to the main steps.

**Setting the constants.** Given \( \gamma \), we set \( \alpha := \left(\frac{\gamma}{2}\right)^4 \); this will be our approximation factor for the embedding of \( T - Y \). We choose \( \varepsilon \) such that

\[
\varepsilon \leq \gamma^{10}.
\]

(32)

Apply Lemma 4.4 (the regularity lemma) to \( \varepsilon^2 \) and \( \gamma := \frac{1}{\varepsilon} \) obtaining numbers \( M_1 \) and \( n_0 \). We choose \( \beta \ll \varepsilon \), and \( \lambda \leq \frac{\beta^2 - \varepsilon}{3000} \). Finally, choose a sufficiently large number \( m_0 \) for the output of Lemma 7.3. Resumingly, we have

\[
\frac{1}{m_0} \ll \lambda \ll \beta \ll \varepsilon \ll \alpha \ll \gamma.
\]

(33)

**Holes.** Call a subset of \( V(G) \) containing at least \( \gamma m \) vertices a hole if it induces less than \( 100\varepsilon m^2 \) edges. Let \( V_{\text{Bad}} \) be the set of all vertices in \( V(G) \) whose nonneighbourhood contains a hole. For now, suppose that

\[
|V_{\text{Bad}}| \leq \frac{m}{2}.
\]

(34)

(The other case will be treated at the end.) Now, assume we are given a graph \( G \) as in the lemma, a set \( S \), a tree \( T \) and a subtree \( T^* \) of \( T \), with

\[
|V(T^*)| < \gamma m,
\]

(35)
a vertex \( t^* \in V(T^*) \), and a set \( Y \subseteq V(T^*) \setminus \{t^*\} \). By (34), we can choose \( W := V(G) \setminus V_{\text{Bad}} \). Then there is an embedding \( \varphi \) of \( T^* - Y \) into \( G - S \) with

\[
\varphi(t^*) \notin V_{\text{Bad}}.
\]

(36)

**Regularising the host graph.** We take an \( \varepsilon^2 \)-regular partition of \( G' := G \setminus (\varphi(V(T^* - Y) \cup S) \), with a reduced graph \( R_G \), on \( M_0 \leq p' \leq M_1 \) vertices. We wish to extend the embedding of \( T^* - Y \) to an embedding of all of \( T - Y \) into \( G - S \). Note that the minimum degree of \( R_G \), is no longer bounded from below by \( \left(\frac{2}{3} - 13\varepsilon\right)p \), as in the proof of Lemma 2.1, because of the possible degree into the set \( \varphi(V(T^* - Y)) \). But we can guarantee the following bound:
\[
\delta_{w(R_{G,\gamma})} \geq \left(\frac{2}{3} - \gamma + \left(\frac{\gamma}{2}\right)^4 - 13\varepsilon\right)p' \geq \left(\frac{2}{3} - \gamma + \frac{\gamma^4}{20}\right)p'.
\] (37)

Cutting the tree. We use Lemma 5.1 to cut up the tree induced by \( V(T - T^*) \cup \{t^*\} \), making \( t^* \) a seed. Add all neighbours of \( t^* \) belonging to a tree from \( F_2 \) to the set \( W \) of seeds. Use Lemma 5.1(f)–(h) to see that there are at most \( 2\beta(m - |V(T^*)|) \) new seeds (with exactly the same argument as the one used to prove Lemma 5.1(i)). Note that the new seeds transform the partition of the tree a little, as any new seed cuts the tree from \( F_2 \) it belonged to. We just add the newly formed small trees to \( L, F_1, F_2 \setminus F_2' \) as appropriate, and, slightly abusing notation, continue to call these sets \( L, F_1, F_2 \setminus F_2' \) or \( F_2' \). Let \( W^* \) be the set of all seeds adjacent to \( t^* \). It will not be necessary to add any more extra seeds, as we did in the proof of Lemma 2.1, so the total number of seeds is bounded by \( \frac{3}{\beta^2} \).

Embedding leaves incident with \( t^* \), and reserving for \( W^* \). First embed the leaves incident with \( t^* \), into any cluster, using the minimum degree of \( G \). As \( t^* \) has at most \( \lambda m \ll \varepsilon m \) leaves hanging from it, this will not disturb the rest of the embedding process. Let \( L' \) denote the set of the remaining leaves from \( L \).

Next, we choose a cluster \( C^* \) such that at least a third of its vertices are neighbours of \( \varphi(t^*) \). We reserve a set \( C_{W}^* \subseteq C^* \) of size \( \varepsilon^3 m \) consisting of neighbours of \( \varphi(t^*) \) in \( C^* \). This reservation ensures that we will not block the neighbourhood of \( \varphi(t^*) \) before embedding \( W^* \).

Embedding the trees from \( F_1^* \). Now we embed the trees from \( F_1^* := F_1 \cap T_{t^*} \). We provisionally slice up each of the clusters of \( R_{G,\gamma} \), into \( \frac{1}{\varepsilon} \) smaller sets (slices) of equal sizes (plus a very small garbage set), in a way that at most one of the new slices contains both neighbours and nonneighbours of \( \varphi(t^*) \). Let \( S \) be the set of all these slices except for the garbage set and the mixed slice, and let \( R_{G,\gamma} \), be the reduced graph on \( S \). Say \( |S| = p'' \). Note that \( R_{G,\gamma}' \) is still regular (with a slightly worse approximation), and in \( R_{G,\gamma}' \), the minimum degree condition from (37) becomes

\[
\delta_{w(R_{G,\gamma}')} \geq \left(\frac{2}{3} - \gamma\right)p''.
\] (38)

Consider a set \( N \subseteq S \) of size \( \left(\frac{2}{3} - \gamma\right)p'' \) such that \( \varphi(t^*) \) is adjacent to all vertices in all clusters of \( N \). We will now find a matching \( M^* \) and path partitions \( P_A^*, P_B^* \), or slight variations thereof, as in the proof of Lemma 2.1, where we employed Lemmas 4.2 and 4.3.

Let us start with the matching \( M^* \) from Lemma 4.2. While the conditions of the lemma are still satisfied if we take \( \xi \) of the order of \( \gamma \), we would like to have an outcome with \( \xi \) having the order of \( \varepsilon \). The only possible reason that could prevent us from finding a set \( Y \) of order around \( 500\varepsilon p'' \) and an \( N \)-good perfect matching \( M^* \) of \( R_{G,\gamma}' \), \( Y \) as in Lemma 4.2 is that the first line of the proof of Lemma 4.2, where we greedily match using the minimum degree condition, fails in our new circumstances (for the rest of the argument in that proof we only need a much weaker minimum degree condition). So, if we cannot find \( M^* \), then each matching from \( V(R_{G,\gamma}') \setminus N \) to \( N \) leaves more than \( \lfloor 500\varepsilon p'' \rfloor \) vertices from \( V(R_{G,\gamma}') \setminus N \) uncovered. Thus there is a Hall-obstruction, that is, a set \( X'_1 \subseteq V(R_{G,\gamma}') \setminus N \) with less than \( X'_1 \mid \lfloor 500\varepsilon p'' \rfloor \) neighbours in \( N \). Now, apply (38) to any vertex from \( X'_1 \), and set \( X_3 := N(X'_1) \cap N \) and \( X_i := V(R_{G,\gamma}') \setminus N \) to see that
\[ X_i \cup X_j \geq \left( \frac{2}{3} - \gamma \right)p'^{n}. \tag{39} \]

Moreover, as \( |X_i| = p'' - |N| \), we see that
\[
\left( \frac{1}{3} + \gamma \right)p'' \geq |X_i| \geq \left( \frac{1}{3} - \frac{\gamma}{2} + 250e \right)p'' \geq \left( \frac{1}{3} - \frac{\gamma}{2} \right)p''.	ag{40}
\]

where for the second inequality we use (39), as well as the fact that \( X_i < |X_i| - [500ep'' \leq |X_i| - [500ep'' \), which implies that
\[
|X_i| \geq |X_i \cup X_j| - |X_i| \geq \frac{|X_i \cup X_j|}{2} + 250ep''.
\]

Using (39) and (40), we can bound the size of \( X_3 \) as follows:
\[
\left( \frac{1}{3} + \gamma \right)p'' \geq |X_i| = |X_i \cup X_j| - |X_i| \geq \left( \frac{1}{3} - 2\gamma \right)p''. \tag{41}
\]

By (40) and (41), and by the choice of \( X_i' \), we also have
\[
\left( \frac{1}{3} + \gamma \right)p'' \geq |X_i| \geq |X_i'| \geq \left( \frac{1}{3} - 2\gamma \right)p''. \tag{42}
\]

In particular, the fact that \( |X_i'| \geq |X_j| \) together with (40) ensures that \( X_i := X_3 \cup (X_i \setminus X_i') \) has a size at most \( \left( \frac{1}{3} + \gamma \right)p'' \), and hence, by (41),
\[
\left( \frac{1}{3} + \gamma \right)p'' \geq |X_i'| \geq |X_j| \geq \left( \frac{1}{3} - 2\gamma \right)p''.	ag{43}
\]

Letting \( X_2 \) denote the nonneighbours of \( X_i' \) in \( N \), we obtain from (38) in a similar way as for \( X_i \) that \( |X_2 \cup X_j| \geq \left( \frac{2}{3} - \gamma \right)p'' \), and therefore, using (41) for a bound on \( |X_j| \), we obtain that
\[
|X_2| = |X_2 \cup X_j| - |X_j| \geq \left( \frac{1}{3} - 2\gamma \right)p''. \tag{44}
\]

Also, by (39), we have that
\[
|X_2| \leq \left( \frac{1}{3} + \gamma \right)p''. \tag{45}
\]

But then \( G \) is \( \gamma \)-special. Indeed, (35), (42), (43), (44) and (45) imply that Definition 7.1(i) holds for \( \bigcup X_i' \), \( \bigcup X_2 \) and \( \bigcup X_i' \cup \phi(V(T^*) - Y) \cup S \cup (V(G') \setminus \bigcup_{S \in S} V(S)) \), while Definition 7.1(ii) holds by the definition of \( X_2 \), and since by (32), any nonedge in \( R'_G \), corresponds to a very sparse pair of clusters in \( G \). However, \( G \) being \( \gamma \)-special is against
the assumptions of Lemma 7.3. So, no Hall-type obstruction can exist, and we find the matching $M^*$ as desired. Set $Q := V(M^*) \setminus N$.

Next, we turn to the good path partitions $P^*_Q$ and $P^*_R$ from Lemma 4.3. Let us start with $P^*_Q$. In the proof of Lemma 4.3, we constructed $P^*_Q$ using $M^*$ and an auxiliary matching $M^Q$, which, in turn, was obtained as the union of two matchings, the first of which is $\tilde{M}^Q$, a maximum matching inside $V(M^*) \setminus N$. We define $\tilde{M}^Q$ in the same way here, and set $\tilde{Q} := V(M^* \setminus (N \cup V(\tilde{M}^Q)))$. Note that $V(\tilde{M}^Q) \geq 6\gamma n$, as otherwise, $\tilde{Q}$ is a hole, which is impossible by (36). In the proof of Lemma 4.3, $\tilde{M}^Q$ is completed to $M^Q$ by matching clusters from $\tilde{Q}$ to clusters in $N \setminus R$, where $R$ consists of the clusters matched to $\tilde{Q}$ by $M^*$. By the maximality of $\tilde{M}^Q$, $\tilde{Q}$ is independent, and all but at most one of the vertices in $\tilde{Q}$ see at most one of the endvertices of any edge in $\tilde{M}^Q$. So, these clusters see at least $|N \setminus R| - \frac{|V(\tilde{M}^Q)|}{2} \geq |\tilde{Q} \setminus V(\tilde{M}^Q)|$ clusters in $N \setminus R$, implying that we can match all but at most one vertex of $\tilde{Q}$ in $M^Q$, and thus find an $N$-in-good path partition $P^*_Q$, as desired.

Let us now turn to the $N$-out-good path partition $P^*_R$ from Lemma 4.3. To find $P^*_R$ we employed a maximum matching $\tilde{M}^R$ inside $R$. If $|V(\tilde{M}^R)| > 6\gamma n$, we can proceed as above to find $P^*_R$. Otherwise, it is not hard to see that we can match all except at most $6\gamma n$ clusters of $R$ to $N \setminus R$, which leads to almost all thus obtained paths having six vertices. We can then match the remaining $6\gamma n$ clusters of $R$ to the innermost vertices of these paths. In this way, we find a partition $P^*_{\text{mod}}$ of $R$ into vertex-disjoint copies of graphs of the following types: any of the three types of paths allowed in an $N$-out-good path partition, plus the graph $G_{\text{good}}$, which is defined by $V(G_{\text{good}}) = \{A, A', B, B', C, D, E, E', F, F', \}$, with $B, B', C, D, E, E' \in N$, and $E(G_{\text{good}}) = \{AB, A'B', BC, B'C, CD, DE, DE', EF, E'F'\}$.

We embed $\cup F^*_i$ using $M^*, P^*_Q$ and $P^*_{\text{mod}}$, all the time avoiding $C_{W^*}$. Note that when using $P^*_{\text{mod}}$ we will have to adapt our strategy from Section 6.3.2, but this is not hard. To leave all clusters balanced during the embedding of $\cup F^*_i$, we again use a small set $U'$ of pseudo-used vertices.

_Slicing up the clusters._ We now go back to work in $R_G$. We slice up the yet unused parts of the clusters as before (but avoiding $C_{W^*}$), into sets $C_A, C_W, C_V, C_{F_1 \setminus F^*_i}$ and $C_{F_2}$. The slices $C_X$ reflect the sizes of the corresponding sets $X$, but we leave sufficient buffer space in each.

We now go through the subtree induced by $W$ and the nontrivial trees hanging from them in a connected way, starting with the root $t^*$. As before, we embed each seed together with all small trees from $F_1 \cup F_2$ hanging from it. We always avoid the set $C_W$, unless we are embedding a seed from $W^*$.

---

6For any graph of the new type, consider embedding $x$ trees of type $T$ in the edge $AB$, putting the bipartition class $T_1$ that contains the root $r_T$ into $B$ and the other class $T_2$ into $A$. Then embed $x$ trees similarly into each of $A'B', E'F'$ and $E'F''$. Next, do $y$ rounds of embedding $T$ with $r_T$ going into $C, T_2$ going to $B$, and $T_1 - r_T$ going to $A$, and $y$ analogous rounds for each of the edges $A'B', E$ and $E'F'$. Finally, embed $T$ $2z$ times into $CD$, of which $z$ rounds in each “direction.” After these $4x + 4y + 2z$ rounds, we filled each of $A, A', F', F''$ with $x_1 + y(t_1 - 1)$ vertices, each of $B', B', E, E'$ with $x_2 + y(t_2 - 1)$ vertices, and each of $C, D, E$ with $2y + zt$ vertices, where $t_i = |T_i|$ for $i = 1, 2$. So, taking $x = (t_1 + t_2) \cdot (t_1 - t_1 + 1), y = (t_1 + t_2) \cdot (t_2 - t_2)$ and $z = (t_1 + t_2 - 2) \cdot (t_2 - t_2) + t_2$, we fill each of the eight clusters with exactly the same amount of vertices. Similarly as in Section 6.3.2, we can calculate how to fill all types of graphs from $P^*_{\text{mod}}$ simultaneously with the same amount of vertices.
Embedding a seed $s$. We embed each seed $s$ in a neighbour $\varphi(s)$ of the image $\varphi(p)$ of its parent $p$, with $\varphi(s) \notin V_{bad}$, and such that $\varphi(s)$ is typical with respect to the slices $C_L, C_W, C_V$ and $C_{F_1 \setminus F_2}$. Note that this is possible as by (34), and by our condition on the minimum degree of $G$, the vertex $\varphi(p)$ has plenty of neighbours outside $V_{bad}$. Usually, seeds go to $C_W$, but seeds from $W^*$ go to their reserved space $C_{F_2}$. Note that we did not need to group and order our seeds as in the proof of Lemma 2.1, and we also do not need to choose the target clusters as carefully.

Embedding the trees from $F_2 \cup F_1$ at $s$. For the trees from $F_2$, we can find the matching $M_{F_1}$ just as before, as the minimum degree bound (38) is sufficient. Again, we make the connections through the slices $C_V$, using the fact that the corresponding seeds are embedded in typical vertices (with respect to almost all slices $C_V$).

For the trees from $F_1$, we proceed as above for $F_1^*$, that is, we slice up the clusters, so that almost all of the obtained slices behave uniformly with respect to being adjacent to $\varphi(s)$. Let $R_G^*$ be the graph on these slices, after momentarily discarding the mixed slice and the garbage slice. Since $G$ is not $\gamma$-special, and since we embedded $s$ outside any hole, we can proceed exactly as above (when we embedded the trees from $F_1^*$) to find a matching $M$, an in-good path partition and a modified out-good path partition in $R_G^*$.

Embedding the leaves from $L'$. In (33), we chose $\lambda$ such that
\[ L' \leq \frac{3}{\beta^2} \cdot \lambda m \leq \frac{1}{1000} \epsilon m. \]

Since there are about $p'$ slices $C_L$, each much larger than $\epsilon_p^m$, and since the embedded seeds (except possibly $t^*$) see about two-thirds of almost all slices $C_L$, we can embed the leaves from $L'$ greedily into $\bigcup_{C \in V(R_G^*)} C_L$. This finishes the proof of Lemma 7.3 for the case that (34) holds.

More holes. Now assume that (34) does not hold. We can delete a few vertices from a hole so that inside the remaining set, no vertex has a degree more than $\sqrt{\epsilon} m$. By deleting a few more vertices, we arrive at a hole of size exactly $\left\lfloor \left( \frac{1}{3} - 8\gamma \right) m \right\rfloor$. Let $\mathcal{H}$ be the set of all holes obtained in this way. As $G$ has a minimum degree at least $\left\lfloor \frac{2m}{3} \right\rfloor$ for each $H \in \mathcal{H}$ and $x \in V(H)$ we have
\[ \deg(x, V(G) \setminus V(H)) \geq |V(G) \setminus V(H)| - 9\gamma m. \] (46)

In particular, since (34) is not true, $\mathcal{H}$ contains at least two holes, $H_1$ and $H_2$, and their intersection is empty.

By (46) it is easy to calculate that for $i = 1, 2$, all but at most $3\sqrt{\gamma} m$ vertices of $V(G) \setminus (H_1 \cup H_2)$ have a degree at least $|H_i| - \sqrt{\gamma} m$ into $H_i$. We choose a set $H_3 \subseteq V(G) \setminus (H_1 \cup H_2)$ with $|H_3| = |H_1| = |H_2|$ and such that for each $x \in H_3$, and each $i \in \{1, 2\}$, $\deg(x, H_i) \geq |H_i| - \sqrt{\gamma} m$.

Set $H_0 := V(G) \setminus (H_1 \cup H_2 \cup H_3)$. As $\delta(G) \geq \left\lfloor \frac{2m}{3} \right\rfloor$, each vertex in $H_0$ sees at least two-fifths of the vertices at least two of the sets $H_1, H_2, H_3$. So, there is an index $i \in \{1, 2, 3\}$, without loss of generality let us assume $i = 2$, such that at least two-thirds of the vertices in $H_0$ are adjacent to at least two-fifths of the vertices in $H_2$. In other words, we can split $H_0$ into two sets $H_4, H_5$ as in Lemma 7.5.
Let $T_{\text{Bad}}$ be the set of all bad trees in $F_1$. If $|T_{\text{Bad}}| \geq \frac{33}{100} m$, we add the trees from $T^* - t^*$ to $L$, $T_{\text{Bad}}$, $F_1 \setminus T_{\text{Bad}}$, and $F_2$, where we now allow that trees in $F_2$ have more than $\beta m$ vertices. Forgetting about the embedding $\phi$ of $T^*$, we apply Lemma 7.5 to $G$, finding that $T \subseteq G$, contrary to the assumptions of Lemma 7.3. So,

$$|T_{\text{Bad}}| \leq \frac{33}{100} m.$$ 

Now, we let $t^*$ be embedded into the hole $H_1$. We regularise $G - \phi(T^* - Y)$, respecting the prepartition given by $H_1, H_2, V(G) \setminus (H_1 \cup H_2)$. In the reduced graph $R_G$ on $p$ clusters, we use (46) to find an edge $X_i X_2$ with $X_i \subseteq H_i$ for $i = 1, 2$, and $|X_2 \cap N(\phi(t^*))| \geq \frac{|X_2|}{2}$. Using (46) again, we find a set $D$ of disjoint triangles $ABC$ in $R$ with $A \subseteq H_1, B \subseteq H_2$ and $C \subseteq H_3$, such that for each $i = 1, 2$, two of the pairs $(A, X_i), (B, X_i), (C, X_i)$ have large density (larger than $\frac{2}{3}$), and such that moreover, $\phi(t^*)$ sees at least $\frac{2}{3}$ of the vertices in $B$ and in $C$. We can choose $D$ such that $|D| = \left\lceil \left(\frac{1}{3} - 100 \gamma\right)p \right\rceil$.

We now split the clusters from $V(R_G) \setminus \{X_1, X_2\}$ into slices. Each cluster belonging to a triangle from $D$ is split into one slice $D_1$ of size $\frac{|T_{\text{Bad}}|}{|D|} + 100 \epsilon m$, and one slice $D_2$ containing the remaining vertices. Doing this, we put as many neighbours of $\phi(t^*)$ as possible into $D_1$. Let $S$ be the set of all slices of type $D_1$ and note that together they are large enough to accommodate all trees from $T_{\text{Bad}}$ (there is even some buffer space). Let $D'$ denote the set of triangles in the reduced graph on $S$ corresponding to triangles from $D$. Now, take the set of all slices of type $D_2$, and all clusters in $V(R_G) \setminus (V(D) \cup \{X_1, X_2\})$, and split each of these elements into slices of size $\epsilon m$ (plus possibly one garbage set). Let $S'$ be the set of all these slices, and denote by $R_G'$ the reduced graph on $S'$. By our bound on $|T_{\text{Bad}}|$, we know that

$$\delta(R_G') \geq \left(\frac{2}{3} - 500 \gamma\right)p',$$  \hspace{1cm} (47)

where $p' = |S'|$. Now, we embed the leaves hanging from $t^*$ as before, and embed the bad trees at $t^*$ into $\bigcup S'$, using regularity inside the triangles, and filling the clusters from $S'$ as evenly as possible. For this, consider a triangle $A'B'C' \in D'$ and note that $\phi(t^*)$ sees almost $\frac{2}{3}$ of the clusters $B'$ and $C'$. So, if necessary we can fill almost all $A' \cup B' \cup C'$ with trees from $T_{\text{Bad}}$, by distributing their first vertices among $B'$ and $C'$.

We then embed the trees from $F_1 \setminus T_{\text{Bad}}$ adjacent to $t^*$. For this, we temporarily slice up the clusters from $S'$ into about $\frac{1}{\epsilon}$ new slices, so that almost all new slices contain either only neighbours or only nonneighbours of $\phi(t^*)$, disregarding the possible garbage slice and the possible mixed slice, as before. Call this new set of slices $S''$.

In the reduced graph on $S''$, we find a matching $M^*$ (for this, we use Equation 47 and the fact that most of the vertices in the nonneighbourhood of $\phi(t^*)$ lie in $H_1$, which means that they see almost all of $V(G) \setminus H_1$). We then find a modified out-path partition $P'_{\text{mod}}$ as before. We may be unable to find an in-good path partition $P_0$ as before (as now $\phi(t^*)$ lies in a hole). But, in a similar way as we found $P'_{\text{mod}}$, we can find a modified partition $P''_{\text{mod}}$, which allows for graphs of the three types from the definition of the in-good path partition, plus a new type
of graph on eight vertices $A, A', B, B', C, D, E, E'$, with $A, A', C, D, F, F' \in N$, and
dges $AB, A'B', BC, B'C, CD, DE, DE', EF, E'F'$. Recall that the in-good path partition was
only used for the trees from $T_{NearBal}$. It is not hard to see that all trees in $T_{NearBal} \setminus T_{Bad}$ can be
embedded into the modified in-good path partition $\mathcal{P}'_{mod}$. We fill all clusters from $\bigcup S'$
evenly (again using a set $U'$).

Next, we slice up the yet unused parts of the clusters as before into sets $C_i, C_W, C_V, C_F \setminus F', C_F$, each of the size needed, plus buffer space. We go through the
seeds $s$ and the trees from $F_1 \cup F_2$ hanging from $s$. We embed $s$ into a vertex $\varphi(s) \in X_i$, for
some $i = 1, 2$, such that $\varphi(s)$ is typical with respect to each of the clusters from
triangles $A'B'C' \in \mathcal{D}'$, with respect to the unused parts of these clusters, with respect to
the unused parts of slices of clusters from $S'$, and with respect to $X_{3-i}$. We also require
that $s$ is embedded into a neighbour of the image of its parent, which is not a problem,
since all such parents are embedded into vertices having sufficient degree into $X_1 \cup X_2$.

We embed all trees from $T_{Bad}$ hanging from $s$ into $\bigcup S$, using regularity and filling all
clusters from $S$ almost evenly. Note that $\varphi(s)$ is typical with respect to the unused part of
at least two of the clusters of any triangle $A'B'C' \in \mathcal{D}'$, that is, $\varphi(s)$ sees almost $\frac{2}{3}$ of these
unused parts. So, as above for $\varphi(t^*)$, we can distribute the first vertices of the trees in $T_{Bad}$
at $s$ among the two clusters and thus fill up the triangle if necessary.

We then embed the trees from $F_1 \setminus T_{Bad}$ at $s$ into $\bigcup S'$, using the same strategy as before,
that is, temporarily slicing up the clusters from $S'$ into about $\frac{1}{\xi}$ new slices, so that almost
all contain either only neighbours or only nonneighbours of $\varphi(s)$. Finding a matching $M^*$
and modified path partitions as above, we can fill all clusters from $\bigcup S'$ evenly. For the
trees in $F_2$ hanging from $s$, we use a matching $M_{F_2}$, as before. If $T \in F_2$ contains a parent
$p$ of a seed, we embed $p$ into a suitable vertex $v \in C_V$ with many neighbours in $X_1 \cup X_2$.
We finish by embedding the leaves hanging from seeds as before.

It remains to prove Lemma 7.5. For its proof it will be convenient to have the following
auxiliary result at hand.

**Lemma 7.6.** Let $H$ be a graph with universal vertex $w$. Let $H_1, H_2, H_3 \subseteq V(H)$ be disjoint,
with $|H_i| \geq \frac{9}{10}m$ and such that each vertex of $H_i$ is adjacent to at least $\frac{9}{10}$ of the vertices
in $H_i$, for $i = 1, 2, 3$ and $j \in \{1, 2, 3\} \setminus \{i\}$. Let $T$ be a tree on $m + 1$ vertices, with sets
$W, L, F_1, F_2, F_3$ as in Lemma 5.1 for some $\beta < 1$ with $\beta^2m > 600$, except that we do not
require the upper bound in Lemma 5.1(f). If $F_1$ contains at least $\frac{33}{100}m$ bad trees, then there is
a tree $T' \subseteq T$ with $|V(T')| \leq \frac{m}{50}$ and an embedding $\varphi$ of $T'$ into $H$ such that

(i) each component of $T - T'$ is a bad tree from $F_1$, and for at least $\frac{33}{400}m$ of these trees,
their neighbour in $\varphi(T')$ is embedded in $H_2 \cup \{w\}$ and

---

7To show this, let us just prove that any graph of the new type can be equally filled with a fixed type of tree
$T \in T_{NearBal} \setminus T_{Bad}$. For this, let $t$ denote the size of the larger partition class of $T$. Note that $t \geq 3$ and the other class has
size $t - 1$. Let $r_T$ be the root of $T$. Embed $2t - 1$ times $r_T$ into $C$, the smaller bipartition class into $B$, and the rest into
$A$. Do the same three more times, replacing the triple $C, B, A$ with $C, B', A'$, with $D, E, F$ and with $D, E', F'$. Embed
$t - 3$ times the larger bipartition class into $C$, and the smaller class into $D$. Also embed $t - 3$ copies of $T$ the other way
around into $D$ and $C$. Then each of the eight clusters is filled with exactly $(2t - 1)(t - 1)$ vertices.
(ii) \( |\varphi(V(T')) \cap H_2| \geq |\varphi(V(T')) \cap H_1| \) for \( i = 1, 3 \).

**Proof.** Step 1: Preparation. Let \( T_{\text{Bad}} \) be the set of all bad trees in \( F_1 \) and let \( T'' \) be the union of \( W \) and all trees from \( (L \cup F_1 \cup F_2) \setminus T_{\text{Bad}} \). Note that by the assumptions of the lemma,

\[
|T''| \leq \frac{m}{100}.
\]

(48)

Consider the bipartition \( W_0 \cup W_1 \) of \( W \) induced by the bipartition of \( T \) and say \( W_0 \) is adjacent to at least as many trees from \( T_{\text{Bad}} \) as \( W_1 \). So, there are at least \( \frac{33}{200} m \) bad trees hanging from vertices of \( W_0 \).

Partition \( W_0 \) into three sets \( W'_0, W'_2, W'_3 \) such that \( W'_0 \) contains all seeds from \( W_0 \) not adjacent to any leaves, and such that \( |(\ell_{W'_1} + |W'_2|) - (\ell_{W'_2} + |W'_3|)| \) is minimised, where we define \( \ell_X \) as the number of leaves adjacent to the vertices of \( X \), for any \( X \subseteq W \). Say \( W'_2 \) is adjacent to at least as many bad trees as \( W'_3 \). If \( \ell_{W'_1} + |W'_2| \geq \ell_{W'_2} + |W'_3| \), choose \( s^* \) arbitrarily in \( W'_3 \), and if \( \ell_{W'_1} + |W'_2| < \ell_{W'_2} + |W'_3| \), there is at least one vertex in \( W'_2 \) adjacent to more than one leaf, let \( s^* \) be such a vertex. Then, set \( W'_2 := (W'_0 \cup W'_2) \setminus \{s^*\} \) and \( W'_3 := W'_3 \setminus \{s^*\} \), and note that by construction,

\[
\text{there are at least } \frac{33}{400} m \text{ bad trees hanging from } W_2 \cup \{s^*\},
\]

(49)

\( W_1 \) and \( W_2 \cup W_3 \) belong to different bipartition classes of \( T \),

(50)

every seed in \( W_3 \) is adjacent to at least one leaf and

\[
\ell_{W'_1} + \ell_{\{s^*\}} + |W'_2| \geq \ell_{W'_2} + |W'_3|.
\]

(51)

Indeed, (52) is obvious if \( s^* \in W'_0 \), and if \( s^* \in W'_2 \), note that by our choice of \( W'_0 \) and \( W'_2 \), we have

\[
\ell_{W'_0} + \ell_{\{s^*\}} + |W'_2\setminus \{s^*\}| > \ell_{W'_2} - \ell_{\{s^*\}} + |W'_3 \cup \{s^*\}|,
\]

which implies (52).

Finally, for each \( s \in W_1 \) that is adjacent to a tree from \( T_{\text{Bad}} \), choose one such tree, and let \( T_1 \) be the set of all these trees. Let \( T' \) be the tree induced by \( T'' \cup T_1 \). Note that since \( |V(T_1)| \leq \frac{6}{p^2} < \frac{m}{100} \) by Lemma 5.1(b), and because of (48), we have that

\[
|T'| \leq \frac{m}{50}.
\]

(53)

Step 2: Embedding \( T' \setminus L \). Now, we will go through all seeds \( s \in W \), starting with any seed and then always choosing a seed whose parent is already embedded. We embed the seed \( s \), and all trees from \( T \in T_{\text{Bad}} \cup (F_1 \setminus (T_{\text{Bad}})) \cup F_2 \) that are hanging from \( s \). (The set \( L \) will be embedded in the next step.)

For each embedded seed, we will ensure that
(a) if $s \in W_i$ then $\varphi(s) \in H_i$ for $i = 1, 2, 3$, and $\varphi(s^*) = w$.

For each tree $T$ embedded in this step, the following will hold:

(b) $|\varphi(V(T)) \cap H_2| \geq \max\{|\varphi(V(T)) \cap H_2|, |\varphi(V(T) \cup s) \cap H_1|\}$ and

(c) if $T$ contains a parent $p$ of a seed $s' \in W_i$, then $\varphi(p) \notin H_i$.

Say in the current step we are dealing with $s \in W$. If $s = s^*$, we embed $s$ into $w$, which is possible since $w$ is universal. If $s \in W_i$, then we embed $s$ into $H_i$. Note that this is possible because of the minimum degree between the sets $H_i$, and by (a) and (50) if the parent is itself a seed, or by (c) if the parent is not a seed. So, (a) holds for $s$. In what follows, our arguments will often use the minimum degree between the sets $H_i$ without explicitly mentioning it.

Now we go through the trees $T \in T_{W_1} \cup (F_1 \setminus (T_{Bad})) \cup F_2$ hanging from $s$, in any order, and embed each of them. Say we are at tree $T$, with root $r_T$. We need to show we can embed $T$ such that (b) and (c) hold. For this, we distinguish between the types of trees in $T_{W_1} \cup (F_1 \setminus (T_{Bad})) \cup F_2$.

Case 1: $T \in T_{W_1}$. In this case, (a) ensures that $\varphi(s) \in H_1$, and so, we can embed the first and the last vertex of $T$ into $H_2$, and the middle vertex into $H_1$, which ensures (b). Note that $T$ contains no parents of seeds, and so (c) is void.

Case 2: $T \in (F_1 \setminus (T_{Bad})) \cup (F_2 \setminus F_2')$. Let $T_1, T_2$ be the partition of $V(T - r_T)$ induced by the bipartition classes of $T$. We can assume that $|T_2| \geq |T_1|$. Additionally, if $|T_1| = |T_2|$, assume $T_2$ is the class containing the neighbours of $r_T$, and choose any vertex $v \in T_1$.

We embed $r_T$ in $H_1$ unless $\varphi(s) \in H_i$, in which case we embed $r_T$ in $H_2$. We embed $T_2$ into $H_2$, and embed $T_1$ into $H_1$, unless $\varphi(s) \in H_i$, in which case we embed $T_1$ into $H_1$. Note that then (b) holds for $T$, unless $|T_1| = |T_2|$ and $\varphi(s) \in H_i$. In that case we move $\varphi(v)$ from $H_i$ to $H_3$, which ensures (b). Note that $T$ contains no parents of seeds, and so, again, (c) is void.

Case 3: $T \in F_2'$. In this case, $T$ contains exactly one parent $p$ of some seed $s'$, say $s' \in W_i$. We choose $T_1, T_2$ as in case 2, and if $|T_1| = |T_2|$ we make sure that $v \neq p$ (this is possible as $T \notin T_{Bad}$ and hence if $|T_2| = |T_1|$, then $|T_2| \geq 2$).

We first try to embed $T$ as in case 2. This is successful (i.e., both (b) and (c) hold) unless $\varphi(p) \in H_i$, so let us assume this is the case. There are now three possibilities: $p = r_T$, $p \in T_1$, $p \in T_2$, which we will treat separately.

Case 3a: $p = r_T$. In this case, by (50), seeds $s$ and $s'$ belong to the same bipartition class of $T$, that is, either both are in $W_i$, or both are in $W_2 \cup W_3$. Recall that in the latter case, the embedding from case 2 maps $r_T$ to $H_i$. So it is impossible that $\varphi(p) \in H_i$, a contradiction.
Case 3b: \( p \in \mathcal{T}_1 \). In this case, we relocate the image of \( p \), moving \( \varphi(p) \) either from \( H_1 \) to \( H_3 \), or from \( H_3 \) to \( H_1 \). If \( \varphi(v) \) was moved from \( H_1 \) to \( H_3 \), we move it back. Now we have an embedding of \( \mathcal{T} \) for which (b) and (c) hold (note that \( \mathcal{T}_2 \) contains at least two vertices).

Case 3c: \( p \in \mathcal{T}_2 \). First assume that \( s \) and \( s' \) lie in distinct bipartition classes of \( T \). Then, as by assumption \( \varphi(p) \in H_2 \) and \( s' \in W_2 \), we have that \( s \in W_1 \), and therefore, by (a), \( \varphi(s) \in H_1 \). If \( |T_2| \leq |T_1| + 1 \), we can embed all of \( \overline{T_1} \cup \{r_1\} \) in \( H_2 \), and all of \( T_2 \) in \( H_3 \). If \( |T_2| \geq |T_1| + 2 \), we can embed \( p \) in \( H_1 \), \( \overline{T_1} \setminus \{p\} \) in \( H_2 \), and \( T_2 \) in \( H_3 \). In either case we have found an embedding fulfilling (b) and (c).

So we can assume that \( s \) and \( s' \) are in the same bipartition class of \( T \). Then \( s \in W_2 \cup W_3 \), and therefore, by (a), \( \varphi(s) \in H_2 \cup H_3 \). Also, \( s \) and \( s' \) being in the same bipartition class of \( T \) means that \( r_T \) and \( p \) are not adjacent. So we can take the embedding from case 2, and move \( \varphi(p) \) from \( H_2 \) to \( H_1 \). This gives an embedding fulfilling (b) and (c).

This finishes the embedding of \( \mathcal{T}' \setminus L \).

Step 3: Embedding \( L \).

Now consider any \( T \in L \), say \( T \) hangs from seed \( s \). If \( s \in W_1 \cup W_3 \cup \{s^*\} \), we embed \( T \) into \( H_2 \), and if \( s \in W_2 \), we embed \( T \) into \( H_3 \).

Step 4: Verifying (i) and (ii). Note that by (a) and by (52), \( |\varphi(W \cup L) \cap H_2| \geq |\varphi(W \cup L) \cap H_3| \), and thus, by (b), \( |\varphi(V(T')) \cap H_2| \geq |\varphi(V(T')) \cap H_3| \). We will show below that

\[
|\varphi(V(T')) \cap H_2| \geq |\varphi(V(T')) \cap H_1|. \tag{54}
\]

Then, by (49), and by (53), we have embedded \( T' \) in a way that both (7.6) and (7.6) hold.

It only remains to see (54). For this, first note that no \( T \in L \) was embedded into \( H_1 \). Furthermore, for each small tree \( T \) from \( T_{W_1} \cup (F_1 \setminus (T_{Bad})) \cup F_2 \), we know the following because of (b): If \( T \) is hanging from a seed \( s \in W_1 \), then \( |\varphi(V(T)) \cap H_2| \geq |\varphi(V(T)) \cup s \cap H_1| \) (observe that the seed \( s \) is included on the right-hand side); and if \( T \) is hanging from a seed \( s \in W_2 \cup W_3 \), then \( |\varphi(V(T)) \cap H_2| \geq |\varphi(V(T)) \cap H_1| \). So, we always embed at least as much into \( H_2 \) as into \( H_1 \), if we, for the moment, disregard possible seeds in \( W_1 \) that have no trees hanging from them. This is still true if we also disregard \( W_2 \) and all of \( L \).

In other words, by letting \( S_0 \) denote the set of all seeds in \( W_1 \) having no trees hanging from them, and letting \( L_3 \subseteq L \) denote the set of all leaves hanging from vertices of \( W_3 \), we know that

\[
|\varphi(V(T') \setminus (W_2 \cup L_3)) \cap H_2| \geq |\varphi(V(T') \setminus S_0) \cap H_1|. 
\]

We finish the proof of (54) by showing that

\[
|\varphi(W_2 \cup L_3) \cap H_2| \geq |\varphi(S_0) \cap H_1|. 
\]

For this, consider any seed \( s \in S_0 \). By Lemma 5.1(c), \( s \) has a child \( s' \in W \). As \( s \in W_1 \), we know that \( s' \in W_2 \cup W_3 \). If \( s' \in W_2 \), then \( \varphi(s') \in H_2 \), which accounts for \( s \), and if \( s' \in W_3 \),
then by (51), $s'$ has a leaf, which was embedded in $H_2$ and which accounts for $s$. Noting that distinct seeds $s$ have distinct children $s'$, we are done.

We are now ready to prove the final ingredient for the proof, Lemma 7.5.

**Proof of Lemma 7.5.** We start by applying Lemma 7.6 to find a tree $T'$ with $|V(T')| \leq \frac{m}{50}$, and an embedding $\varphi$ of it into $H_1 \cup H_2 \cup H_3 \cup \{w\}$ with the properties given in the lemma. In particular, there are at least $\frac{33}{400}m$ unembedded bad trees whose seeds were embedded into $H_2$. We extend $\varphi$ by embedding some of these trees, in a way that we fill all of $H_4 \cup H_5$.

By the assumptions of Lemma 7.5 (and using the minimum degree between the sets $H_i$, and the fact that both $T'$ and $H_4 \cup H_5$ are small), we know that for any $x \in V(H_5)$ we can embed any bad tree $T$ hanging from $x$ mapping its first vertex into $H_1$, its second vertex into an appropriate vertex of $H_3$, and its third vertex into $x$. For any $x \in V(H_4)$, we can put the first vertex of $T$ into either $H_1$ or $H_3$, the second vertex into $H_2$, and the third vertex into $x$. Alternating $H_1$ and $H_3$ when filling $H_4$, and recalling Lemma 7.6(ii), we can ensure that for $i = 1, 3$,

$$|\varphi(T) \cap H_1| \leq |\varphi(T) \cap H_3| \leq |V(T')| + |H_4 \cup H_5| \leq \frac{3}{100}m \quad \text{and}$$

$$|T_{H_i}| \geq \frac{29}{400}m,$$

where $T_{H_i}$ is the set of all yet unembedded trees whose seed is embedded in $H_2$. We now embed some more trees from $T \in T_{H_2}$, in two phases. This time our aim is to balance the sizes of the used parts of the three sets $H_1, H_2, H_3$.

Let $H_i$ be the set containing the least number of embedded vertices. By (55), we know that $i \neq 2$. In the first phase, if $H_i$ has strictly less used vertices than $H_2$, we embed trees from $T_{H_i}$ by putting their first vertex into $H_1$, their second vertex into $H_2$, and their third vertex into $H_i$. We do this until the used parts of $H_1$ and $H_2$ differ by at most one vertex.

In the second phase, we embed trees from $T_{H_2}$, only using $H_1$ and $H_3$, and keeping their used parts almost perfectly balanced, until the one of them has exactly as many used vertices as $H_2$. Then, since the unused part of $G$ is divisible by three (since the unembedded part of $T$ is divisible by three), all three of $H_1, H_2, H_3$ have exactly the same number of used vertices. By (55), and since the used part of $H_2$ has augmented by a factor of at most $\frac{3}{2}$, this number is at most $\frac{m}{20}$.

Moreover, we used at most $\frac{3}{200}m$ trees in the first phase, and at most $\frac{3}{200}m$ trees in the second phase. So by (56), we still have at least $\frac{m}{25} + 2$ yet unembedded trees in $T_{H_i}$. We embed up to two more of these trees, keeping the sets $H_i$ balanced, so that the number of unembedded trees in $T_{H_i}$ is now divisible by 3.

We next embed all yet unembedded bad trees from $T_{\text{Bad}} \setminus T_{H_i}$, distributing their three vertices equally among the three sets $H_i$. Note that the minimum degree between the sets $H_i$ is large enough so that it does not matter that in this step, all but $\frac{m}{25}$ vertices of each $H_i$ may become occupied.
To embed the remaining trees, we will still use the minimum degree between the sets $H_i$ to embed their first and second vertices, and then a Hall-type argument to embed their last vertex. Let us make this more precise. We always embed three paths at a time. The first vertex of the first path goes to $H_3$, and the first vertex of each of the other two paths goes to $H_1$. The second vertex of the second path goes to $H_3$, and the second vertex of the other two paths goes to $H_2$. Doing this for all remaining trees, the unused space $U_i$ in each $H_i$ has the same size, which is at least $\frac{m}{75}$.

Our plan is to embed the third vertex $v_P$ of each path $P$ as follows: $v_P$ goes into $H_3$ if $P$ was a “first path,” into $H_2$ if $P$ was a “second path” and into $H_3$ if $P$ was a “third path.” Let $A_i$ be the set of all images of second vertices on paths whose third vertex is scheduled to go to $H_i$. Using Hall’s theorem we see that if we cannot embed all third vertices as planned, then there is an obstruction, that is, there is an index $i \in \{1, 2, 3\}$ and a set $A' \subseteq A_i$ such that $|N(A') \cap U_i| < |A'|$. In particular, there is a vertex $v \in U_i \setminus N(A')$, and by our minimum degree condition, $v$ is adjacent to all but at most $\frac{m}{250}$ vertices of $A_i$. Hence $|A'| \leq \frac{m}{250} \leq \frac{|A_i|}{2}$. On the other hand, any vertex $a \in A'$ is adjacent to all but at most $\frac{m}{250}$ vertices of $U_i$, and therefore

$$|A'| > |N(A') \cap U_i| \geq \frac{m}{75} - \frac{m}{250} \geq \frac{|A_i|}{2}.$$ 

This finishes the embedding of $T$. \qed

ACKNOWLEDGEMENTS

Research supported by NSERC. Research supported by ANID Regular Grant 1221905, by FAPESP-ANID Investigación Conjunta grant 2019/13364-7, and by ANID PIA CMM FB210005.

ORCID

Maya Stein  http://orcid.org/0000-0001-7922-6413

REFERENCES

1. M. Ajtai, J. Komlós, and E. Szemerédi, *On a conjecture of Loebl*, Graph Theory, Combinatorics, and Algorithms (Kalamazoo, MI, 1992), vol. 1 and 2, Wiley, New York, 1995, pp. 1135–1146.
2. G. Besomi, M. Pavez-Signé, and M. Stein, *Degree conditions for embedding trees*, SIAM J. Discrete Math. 33 (2019), no. 3, 1521–1555.
3. G. Besomi, M. Pavez-Signé, and M. Stein, *Maximum and minimum degree conditions for embedding trees*, SIAM J. Discrete Math. 34 (2020), no. 4, 2108–2123.
4. G. Besomi, M. Pavez-Signé, and M. Stein, *On the Erdős–Sós conjecture for trees with bounded degree*, Combin. Probab. Comput. 30 (2021), no. 5, 741–761.
5. S. Brandt and E. Dobson, *The Erdős–Sós conjecture for graphs of girth 5*, Discrete Math. 150 (1996), 411–414.
6. B. Csaba, I. Levitt, J. Nagy-Győrgy, and E. Szemerédi, *Tight bounds for embedding bounded degree trees*, Fête de Combinatorics and Computer Science (G. O. H. Katona, A. Schrijver, T. Szenyi, and G. Sági, eds.), vol. 20, Springer, Berlin, Heidelberg, 2010.
7. P. Erdős, Z. Füredi, M. Loebl, and V. T. Sós, *Discrepancy of trees*, Studia Sci. Math. Hungar. 30 (1995), no. 1–2, 47–57.
8. P. Erdős and T. Gallai, *On maximal paths and circuits of graphs*, Acta Math. Acad. Sci. Hungar. 10 (1959), no. 3, 337–356.
9. F. Havet, B. Reed, M. Stein, and D. Wood, *A variant of the Erdős–Sós conjecture*, J. Graph Theory. 94 (2020), no. 1, 131–158.

10. P. E. Haxell, *Tree embeddings*, J. Graph Theory. 36 (2001), no. 3, 121–130.

11. J. Hladký, J. Komlós, D. Piguet, M. Simonovits, M. Stein, and E. Szemerédi, *The approximate Loebl–Komlós–Sós conjecture I: The sparse decomposition*, SIAM J. Discrete Math. 31-2 (2017), 945–982.

12. J. Hladký, J. Komlós, D. Piguet, M. Simonovits, M. Stein, and E. Szemerédi, *The approximate Loebl–Komlós–Sós conjecture II: The rough structure of LKS graphs*, SIAM J. Discrete Math. 31-2 (2017), 983–1016.

13. J. Hladký, J. Komlós, D. Piguet, M. Simonovits, M. Stein, and E. Szemerédi, *The approximate Loebl–Komlós–Sós conjecture III: The finer structure of LKS graphs*, SIAM J. Discrete Math. 31-2 (2017), 1017–1071.

14. J. Hladký, J. Komlós, D. Piguet, M. Simonovits, M. Stein, and E. Szemerédi, *The approximate Loebl–Komlós–Sós conjecture IV: Embedding techniques and the proof of the main result*, SIAM J. Discrete Math. 31-2 (2017), 1072–1148.

15. J. Komlós, G. Sárközy, and E. Szemerédi, *Spanning trees in dense graphs*, Combin. Probab. Comput. 5 (2001), 397–416.

16. J. Komlós, A. Shokoufandeh, M. Simonovits, and E. Szemerédi, *The regularity lemma and its applications in graph theory*, Theoretical Aspects of Computer Science, TACSci 2000. Lecture Notes in Computer Science (G. B. Khosrovshahi, A. Shokoufandeh, and A. Shokrollahi, eds.), 2292, Springer, Berlin, Heidelberg, 2002, pp. 84–112. https://doi.org/10.1007/3-540-45878-6_3

17. D. Piguet and M. J. Stein, *An approximate version of the Loebl–Komlós–Sós conjecture*, J. Combin. Theory Ser. B. 102 (2012), no. 1, 102–125.

18. B. Reed and M. Stein, *Embedding spanning trees in graphs of high minimum degree with a universal vertex II: A tight result*, J. Graph Theory. 0 (2022), 0–00. Accepted for publication.

19. V. Rozhoň, *A local approach to the Erdős–Sós conjecture*, SIAM J. Discrete Math. 33 (2019), no. 2, 643–664.

20. J.-F. Saclé and M. Woźniak, *A note on the Erdős–Sós conjecture for graphs without $C_4$*, J. Combin. Theory Ser. B. 70 (1997), no. 2, 229–234.

21. M. Stein, *Kalai’s conjecture for r-partite r-graphs*. European J. Combin. 2019.

**How to cite this article:** B. Reed and M. Stein, *Spanning trees in graphs of high minimum degree with a universal vertex I: An asymptotic result*, J. Graph Theory. 2023;102:737–783. https://doi.org/10.1002/jgt.22897